

IUCIT 4.1

Reference Manual

(By T.Umapathi Anand, JTO New Tech Installation, TN Telecom Circle)
15 APRIL 2006

Table of Contents

1. General Information.....	5
1.1 About IUCIT.....	5
1.2 History of IUCIT.....	5
1.3 What for IUCIT.....	5
1.4 IUCIT in a Nutshell.....	6
1.5 IUCIT Support.....	6
1.6 IUCIT and the Future (TODO)	6
2. Requirements.....	6
2.1 Hardware Requirements.....	6
2.1.1 Minimum Requirement.....	7
2.1.2 Recommended Requirement.....	7
2.2 Software Requirements.....	7
2.3 Personal Skill Requirement.....	7
3. Hardware Setup.....	7
3.1 Linux Server.....	8
3.2 Windows Client.....	8
3.3 LAN Connection.....	9
4. The LAMP/WAMP Configuration.....	9
4.1 Linux.....	10
4.2 Apache Web Server.....	11
4.3 MySQL Database Server.....	11
4.4 PHP.....	12
5. Installation.....	12
5.1 IUCIT on Linux.....	12
5.1.1 Red Hat 9 Linux Installation.....	13
5.1.2 Server Configuration.....	14
5.1.3 RPMs Upgrade.....	14
5.1.4 MySQL User Creation.....	15
5.1.5 IUCIT Source Copy.....	15
5.1.6 Importing The CODES Database.....	16
5.1.7 IUCIT Registration.....	16
5.1.8 Logging into IUCIT.....	17
5.1.9 Managing Profiles.....	17
5.1.10 Creating Site Dependant Code Table	17
5.1.11 Client Configuration	18
5.2 IUCIT on Windows.....	18
5.2.1 Xampp Installation	18
5.2.2 MySQL Installation	18
5.2.3 Starting Services	18
5.2.4 MySQL User Registration	18
5.2.5 IUCIT Source Copy	18
5.2.6 Importing the CODES Database	19
6. IUCIT Links.....	19
6.1 DATA.....	19
6.1.1 Database Explorer.....	19
6.1.2 Create Database.....	19
6.1.3 Create Table.....	19
6.1.4 Copy Table.....	20
6.1.5 Import Text File.....	20
6.1.6 Export Table Data.....	20

6.2 Reconciliation.....	20
6.2.1 Sub Table Creator.....	21
6.2.1.1 Inputs.....	21
6.2.1.2 Background Operations.....	21
6.2.2 Table Comparer.....	21
6.2.2.1 Inputs.....	21
6.2.2.2 Background Operations.....	21
6.2.2.2.1 Data Packing.....	22
6.2.2.2.2 Time Adjustment.....	23
6.2.2.2.3 Time Synchronization.....	23
6.2.2.2.4 Strict Call Matching.....	23
6.2.2.2.5 Redirected Call Matching.....	24
6.2.2.2.6 Split/Slip Call Matching.....	24
6.2.2.2.7 Higher Duration Call Matching.....	25
6.2.2.2.8 Lower Duration Call Matching.....	25
6.2.2.2.9 Optional Rejection.....	25
6.2.2.2.10 Duplicate Identification.....	25
6.2.2.2.11 ISD Call Marking.....	26
6.2.2.3 Valid and Invalid Calls.....	26
6.2.2.4 Errors during Comparison.....	27
6.2.2.5 Intermediate Report.....	27
6.2.3 Verification.....	27
6.2.3.1 Inputs.....	27
6.2.3.2 Working on Temporary Tables.....	28
6.2.4 Super Check.....	28
6.2.4.1 Inputs.....	28
6.2.4.2 Working.....	28
6.2.5 Code Analyser.....	28
6.2.5.1 Inputs.....	29
6.2.5.2 Background Operations.....	29
6.2.5.2.1 Phase(I) Before Feb 2004.....	29
6.2.5.2.2 Phase(II) After Feb 2004.....	29
6.2.6 IUCIT Reporter.....	30
6.2.6.1 Inputs.....	30
6.2.6.2 Types of Reports.....	30
6.2.7 CSV Dump.....	30
6.3 Billing.....	30
6.3.1 Billing Table Creator.....	30
6.3.1.1 Inputs.....	30
6.3.1.2 Background Operations.....	31
6.3.2 Billing code Analyser.....	31
6.3.2.1 Inputs.....	31
6.3.2.2 Background Operations.....	31
6.3.3 Billing Report Generator.....	32
6.3.3.1 Inputs.....	32
6.3.3.2 Working.....	32
6.4 Profiles	32
6.4.1 Show Profiles	32
6.4.2 New Profile.....	32
6.5 Toos.....	33
6.5.1 Data Integrity Analyser	33
6.5.2 Code Table Creator.....	33
6.5.2.1 Inputs.....	33

6.4.1.2 Background Operations.....	33
6.5.3 Distance Finder.....	33
6.5.4 Update XY.....	33
6.5.5 Disk Free Space	34
6.5.6 Sort Table Data	34
6.5.7 Clean Databases	34
6.5.8 And New Code	34
7. Operating Procedure.....	34
7.1 Work with Client.....	35
7.1.1 Data Copy to Client.....	35
7.1.2 Binary to Text Conversion of BSNL Data.....	35
7.1.3 PO Data Formatting.....	35
7.2 Data Upload to Server.....	35
7.3 Work on Server - Reconciliation	35
7.3.1 Data Integrity Check.....	36
7.3.2 Sub Table Creation for BSNL.....	36
7.3.3 Sub Table Creation for Private.....	36
7.3.4 Pre Check Before comparison.....	36
7.3.5 Table Comparison.....	36
7.3.6 Progress Monitoring.....	37
7.3.7 Intermediate Report Analysis.....	37
7.3.8 Data Verification.....	37
7.3.9 Code Analysis.....	37
7.3.10 Report Preparation.....	37
7.4 Work On Server – Billing.....	38
7.4.1 Billing Table Creation	38
7.4.2 Billing Code Analysis	38
7.4.3 Billing Reporter.....	38
8. Problems Faced and Crossed.....	39
9. Support and Contact Information.....	40
Appendix A.....	41
A. IUCIT Standard Table Structure.....	41
B. Values of cat and slab in the code Table.....	42
C. IUCIT Flags.....	43
Appendix B.....	44
A. Important MySQL Commands.....	44

1. General Information

This Document contains most of the useful information about IUCIT, the IUC Reconciliation and Billing software. There are different versions of IUCIT, and this document deals with version 4.1. This is written to explain the reconciling officers about the installation and operating procedures. The background processes have been explained in every stage to know and work effectively.

1.1 About IUCIT

IUCIT is developed by New Tech Installation, Tamil Nadu Telecom Circle to solve the problem of Settling Bills claimed by Private Operators after Reconciliation and also to raise the Bills towards them. This Software is installed and demonstrated by New Tech Installation Staff almost in all major exchanges of Tamil Nadu. The Reconciliation part of this software has been pre validated by a team of Officers from T&D AT, DTAX Maintenance and RMC. STR, Chennai, has validated the Billing Part of this Software for IUC Phase IV.

1.2 The History of IUCIT

IUCIT was first developed at DTAX, Karaikudi, Tamil Nadu by the combined effort of New Tech Installation Staff and DTAX Maintenance staff to settle the Bills claimed by Private Operators. The discrepancies found in the CDRs submitted by the private operators forced for the fast development of CDR Comparison Software. The volume of Data and complication in procedure were the initial threats. But all the hurdles were crossed one by one by proper choice of H/W, S/W and logics. The tireless efforts besides the co-operation extended by the Karaikudi Maintenance Staff, in bringing out this Software, is commendable. After participating a demonstration of about an hour with the accounts staff, GM Karaikudi gave first green signal for the implementation of this Software. After a complete study about the Software, DGM New Tech Installation took this program for Pre Validation. With a good effort from DGM NTI, this software reached all DTAXs and many Local Exchanges of Tamil Nadu. GM NMS, Chennai has given a break through for IUCIT, after analyzing its abilities and made it available to Other Circles.

1.3 What for IUCIT

In the current scenario, it has become extremely necessary to reconcile the Bills claimed by the Private Operators (POs) on Call Data Record (CDR) basis. The Engineering Officers in the field units have been assigned the responsibility of verification of CDRs given by the Private Operators and hence to certify the Bills claimed. BSNL CDR is available in the field and the PO CDRs are received in CDs. The BSNL data is taken from switch in different media based on the type of the switch. The raw data copied from switch is to be converted in to standard text format besides the raw CDR given by the POs. Each and every record given by the PO is to be verified with the BSNL database. The bill may be passed only for those records, which exit in both the databases. The process of comparison is not an easy

task. Lot of problems have been faced and crossed in comparing the CDRs by IUCIT and the same have been discussed in section 8. With IUCIT, it is possible to settle the bills with full confidence since every record passed can be fully justified. In case of Raising Bills towards POs, IUCIT analyses every record of a particular month and categorizes the records and prepares bill in the required format suggested by TRAI. IUCIT 4.1 Supports all the Phases of IUC including the Sixth Amendment.

1.4 IUCIT in a Nutshell

IUCIT compares every record of Private Operators with the records in the BSNL Database and marks the Validity of the record with different flags. Only the valid records are considered in the final report preparation. It generates bills towards Private Operators, based on the BSNL CDR in the format recommended by Regulation Authority.

1.5 IUCIT Support

The site <http://geocities.com/iucit> gives many useful information for the IUCIT Users and it is getting updated every week. The Yahoo groups iucit@yahoogroups.com connects all the users to share their ideas and to receive news letters from the Developers. New Users can join the IUCIT Group at <http://groups.yahoo.com/groups/iucit> to get updated with the latest information and software. Users can download the required tools and update information from <http://geocities.com/umapst/dl.html> and <http://geocities.com/iucit>. Over phone support is available from T.Umapathi Anand (094433-50005), Arul Selvanar (04565-220000).

1.6 IUCIT and the Future (TODO)

- ❖ The Pack is to be made Auto Installable.
- ❖ The Binary to text conversion programs and PO Data formatting programs are to be written in C++ using KDE/Qt to work on Linux.
- ❖ A good structured Client Window is to be created to replace the web browser.
- ❖ The intelligent IUCIT Advisor, negating the manual intervention in the reconciliation process, is to come up.

2. Requirements

2.1 Hardware Requirements

IUCIT Software is designed to have optimum hardware requirements to work with. Depending upon the Volume of data handled and multi user environment, the required configuration may slightly differ from site to site.

2.1.1 Minimum Requirement

- P3/P4 PC with 128 MB RAM, CD Drive and Hard Disk 10GB - 1 No

2.1.2 Recommended Requirement

- PC with 128 MB RAM or more, CD Writer and Hard Disk 80GB - 1 No
 - PC with 32 MB RAM or more, CD Drive and Hard Disk 20GB - 1 No
 - Ether net Cards - 2 Nos
 - X connected LAN Cable (PP) - 5 Mtr
 - Hub (If LAN Support Required) - 1 No
 - Cartridge Drive/ OD Drive / ZipDrive / Audio Tape Drive - 1 No
- as required.

2.2 Software Requirements

IUCIT uses only the freely available but powerful OS and utilities to be economic and efficient. IUCIT can be made to work on Linux or on Windows. Depending upon the choice of OS, the s/w requirement varies as below.

2.2.1 IUCIT on Linux OS

- Red Hat 9 Linux OS Original CDs
- Windows OS (Any version)
- MySQL Front 2.1 (Download from Net)
- RPMs to upgrade Linux (See Appendix for the List of RPMs)
- Alki1.4 /Simi1.7/Dotcy1.0/Rixy1.0 Binary to text converting exes
- IUCIT 4.1 (Available through Yahoo Groups)

2.2.2 IUCIT on Windows OS

- Windows OS (Any version) on NTFS (Recommended)
- xampp-win32-1.4.13-installer.exe
- mysql-4.0.18-win.zip
- Alki1.4 /Simi1.7/Dotcy1.0/Rixy1.0 Binary to text converting exes
- IUCIT 4.1 (Available through Yahoo Groups)

2.3 Personal Skill Requirement

It is very easy to work with IUCIT. Anybody having minimum computer knowledge can process the CDRs using IUCIT. Having Switching Idea, and IUC idea will be more effective. Though it is easy to work with, it may not be so to install and upgrade at present. But nothing is beyond our capacity.

3. Hardware Setup

The following diagram explains you how the hardware is set up for IUCIT with convenient two machines setup.

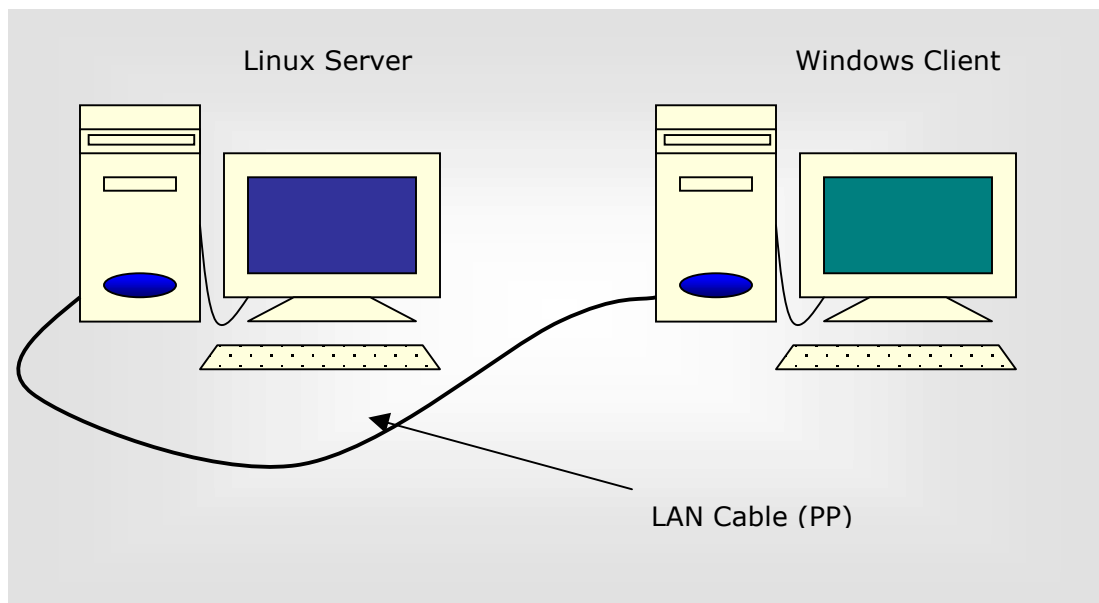


Fig 1

3.1 Linux Server

The Server Machine is storing all the Databases and hence should be of Higher Configuration. Though 64 MB RAM is sufficient to bring up Linux, the experience with GUI will be better if 128 MB RAM. If the volume of Data is less (ie less than 20 Lakhs per Operator) even PIII Machine is sufficient. But in most of the Level II TAXs the figure is about 20 to 50 Lakhs per operator. If time is the prime factor then it is better to shift to P IV. Linux offers secured multi user environment and encloses many servers by default. The LAMP configuration required for IUCIT is easily achieved in Linux. All the Database operations initiated during reconciliation either from Server or from Client is performed in the Linux server.

3.2 Windows Client

The Client machine is normally the billing PC available in any exchange. A PC with cartridge drive for OCB/CDOT, A PC with OD Drive for EWSD, A PC with Zip Drive for AXE and A PC with Audio Tape for 5ESS is used as the client machine. Since Data can be dumped using appropriate drive, the existing billing pc is used as client. In case of unavailability of billing PC, Any PC in the exchange with Ethernet Card can be used as client. But the data has to be brought in CDs to get them processed and uploaded to server. The Binary to text conversion programs developed on VB (at Present) are installed in the client machine and the Exchange Data is converted in to required CSV format. Similarly the data from CDs received from Private Operators are extracted into CSV format text files and these data files are uploaded to server using the MySQL Client software.

3.3 LAN Connection

Normally a Peer-to-Peer connection is set up between the Server and Client. A LAN Cable of sufficient length, cross connected, terminated both ends with RJ45 Connector is used to connect the machines. If the server participates in LAN, then straight connection is made between the machines and the hub/switch. The Server and client should be in the network. Their physical coordinates are immaterial.

4. The LAMP/WAMP Configuration

LAMP Stands for Linux, Apache, MySQL and PHP and WAMP Stands for Windows, Apache, MySQL and PHP. It is a powerful configuration used by IUCIT. The User interactive part (Front End) of IUCIT is Web browser. This software gets input from the users through html forms, gets it processed and gives back the result in html format. The inputs submitted at the Web browser are received by the Apache web server and processed with the PHP Scripts available in the server. Wherever there is PHP tag, the code is sent to PHP Module by Apache and the result is transferred to web server. The PHP Module Processes the codes submitted by Apache and wherever MySQL Query comes, that is sent to MySQL Database server and result is processed and sent to Apache again. The request and report flow are explained in the following diagrams.

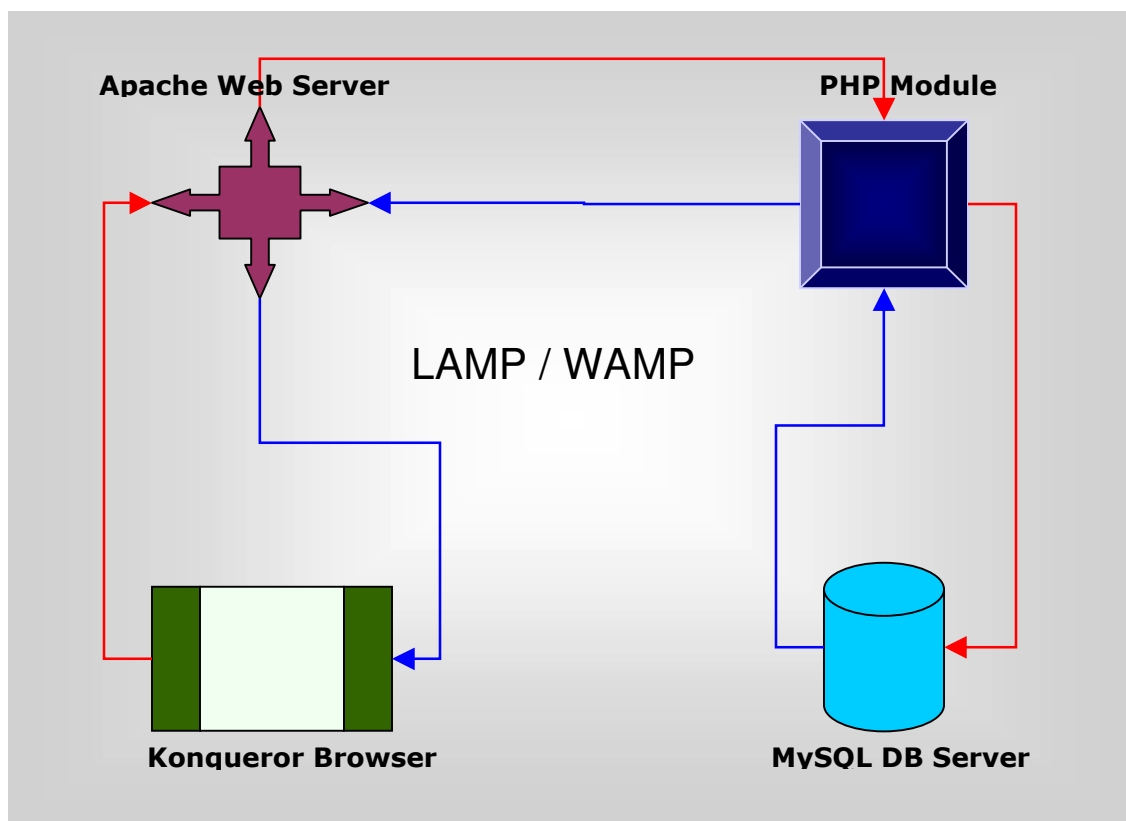


Fig 2

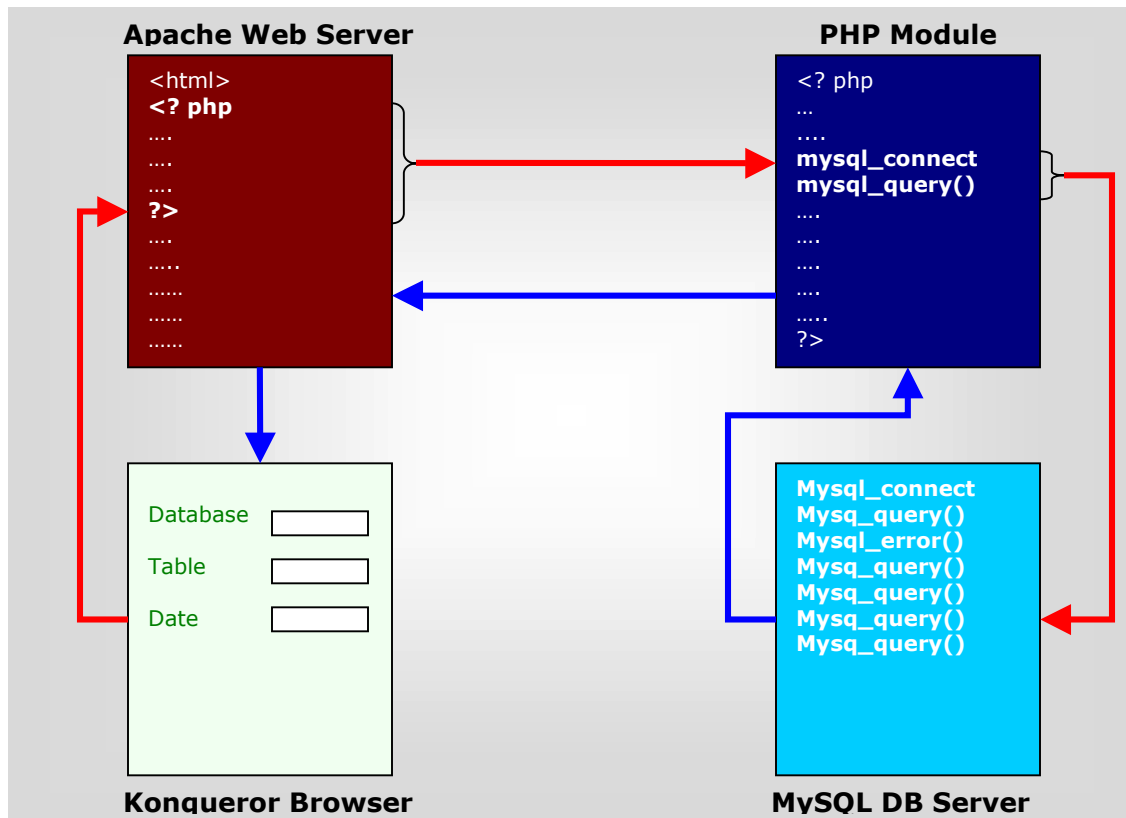


Fig 3

4.1 Linux

Linux is an operating system that was initially created as a hobby by a young student, Linus Torvalds, at the University of Helsinki in Finland. Linus had an interest in Minix, a small UNIX system, and decided to develop a system that exceeded the Minix standards. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released. The kernel, at the heart of all Linux systems, is developed and released under the [GNU General Public License](#) and its source code is freely available to everyone. It is this kernel that forms the base around which a Linux operating system is developed. There are now literally hundreds of companies and organizations and an equal number of individuals that have released their own versions of operating systems based on the Linux kernel. More information on the kernel can be found at our sister site, [LinuxHQ](#) and at the official [Linux Kernel Archives](#). The current full-featured version is 2.6 (released December 2003) and development continues.

Apart from the fact that it's freely distributed, Linux's functionality, adaptability and robustness, has made it the main alternative for proprietary Unix and Microsoft operating systems. IBM, Hewlett-Packard and other giants of the computing world have embraced Linux and support its ongoing development. More than a decade after its initial release, Linux is being adopted worldwide as a server platform primarily. Its use as a home and office desktop operating system is also on the rise. The operating system can also be incorporated directly into microchips in a process

called "embedding" and is increasingly being used this way in appliances and devices.

Throughout most of the 1990's, tech pundits, largely unaware of Linux's potential, dismissed it as a computer hobbyist project, unsuitable for the general public's computing needs. Through the efforts of developers of desktop management systems such as KDE and GNOME, office suite project OpenOffice.org and the Mozilla web browser project, to name only a few, there are now a wide range of applications that run on Linux and it can be used by anyone regardless of his/her knowledge of computers. Those curious to see the capabilities of Linux can download a live CD version called [Knoppix](#). It comes with everything you might need to carry out day-to-day tasks on the computer and it needs no installation. It will run from a CD in a computer capable of booting from the CD drive. Those choosing to continue using Linux can find a variety of versions or "distributions" of Linux that are easy to install, configure and use.

4.2 Apache Web Server

The Apache HTTP Server is a project of the [Apache Software Foundation](#). Apache has been the most popular web server on the Internet since April of 1996. The October 2003 [Netcraft](#) Web Server Survey found that more than 64% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined.

4.3 MySQL Data Base Server

MySQL is a most popular Open Source SQL database, which is very fast, multi-threaded, multi-user Database Server. This is developed, distributed, and supported by MySQL AB, a commercial company, founded by the MySQL developers, that builds its business providing services around the MySQL database. MySQL offers A privilege and password system that is very flexible and secure, and allows host based verification. Passwords are secured because all password transaction are encrypted when you connect to a server. MySQL handles large databases. There are users using MySQL Server with 60,000 tables and about 5,000,000,000 records. Up to 32 indexes per table are allowed. Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 500 bytes (this may be changed when compiling MySQL Server). An index may use a prefix of a CHAR or VARCHAR field. The MySQL web site <http://www.mysql.com/> provides the latest information about the MySQL software. The following table shows the size limit of MySQL database files in different OS.

Operating system	Data File Size
Linux 2.2-Intel 32-bit	2GB (LFS: 4GB)
Linux 2.4+ (using ext3 file system)	4TB
Solaris 9/10	16TB
NetWare w/NSS file system	8TB
Win32 w/ FAT/FAT32	2GB/4GB
Win32 w/ NTFS	2TB
MacOS X w/ HFS+	2TB

Table 1

4.4 PHP

PHP is the acronym for Hypertext Pre Processor. It is a language mostly used for web applications and looks like C/C++. It is very easy to learn and work with. PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. For the majority of the servers PHP has a module, for the others supporting the CGI standard, PHP can work as a CGI processor.

So with PHP, we have the freedom of choosing an operating system and a web server. Furthermore, we also have the choice of using procedural programming or object oriented programming, or a mixture of them. Although not every standard OOP feature is realized in the current version of PHP, many code libraries and large applications (including the PEAR library) are written only using OOP code. PHP is not limited to output only HTML. PHP can output images, PDF files and even Flash movies (using libswf and Ming) generated on the fly. It can output easily any text, such as XHTML and any other XML file. PHP can autogenerate these files, and save them in the file system, instead of printing it out, forming a server-side cache for your dynamic content.

One of the strongest and most significant feature in PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple. The following databases are currently supported:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

5. Installation

IUCIT software is getting upgraded continuously, depending upon the instruction for IUC from TRAI and requirement from the field units. Though not automated, the installation or up gradation of IUCIT at present is easy and user friendly.

5.1 IUCIT on Linux

IUCIT has been installed in most of the exchanges by the New Tech Installation staff in person. The following steps lead to a successful fresh installation of IUCIT Package.

5.1.1 Red Hat 9 Linux Installation

Install Red Hat 9 Linux using the 'Linux Installation' Procedure supplied with the IUCIT Installation Disk or download it from <http://geocities.com/iucit>. Assume that your Server Name is **iuc**, IP Address is **10.0.0.1(255.255.255.0)**, root password is **root123**. Login to Linux as **root** with session option as **KDE**.

5.1.2 Server Configuration

The Server Name and IP address should be set first. The Red Hat symbol displayed at the left bottom of the desktop is like the start button of Windows. Go to Start→System Settings→Network. A window appears as shown in Fig 4. If your NIC(Network Interface Card/ LAN Card) is detected and configured by Linux during Installation, you will find the Device **eth0** in the list of devices shown.

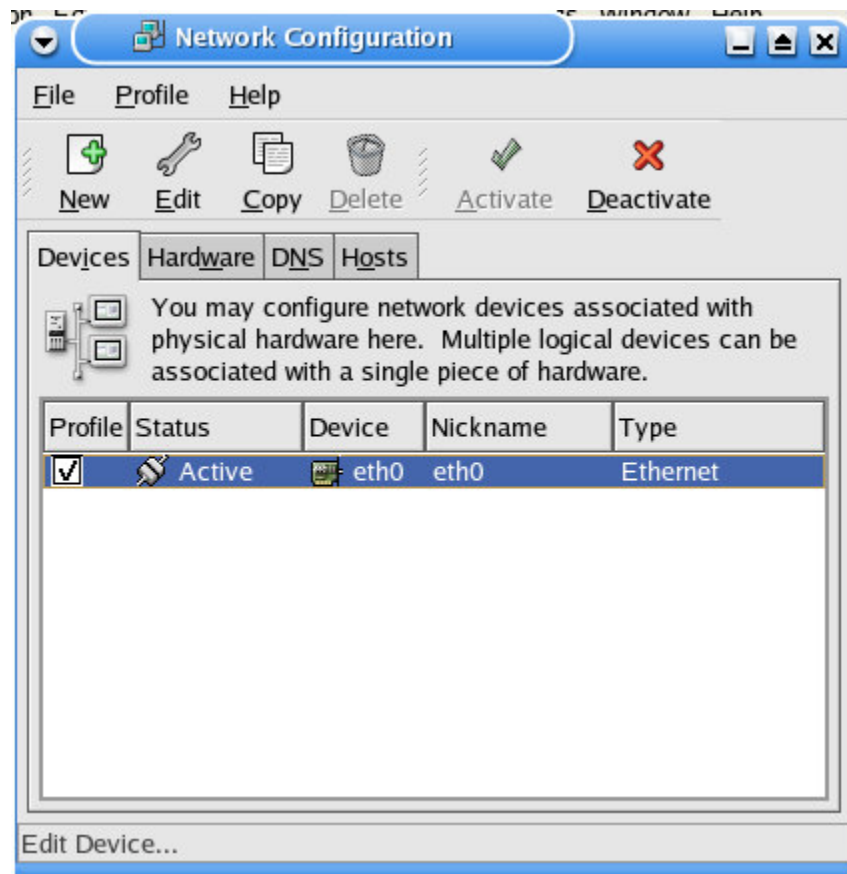


Fig 4

Double Click on the Device to Edit the Values. A window Pop up as Shown in Fig 5. Set the IP address as 10.0.0.1 and Subnet Mask as 255.255.255.0. Give Ok. Select DNS Tab in the Network Window and Set the Host Name as **iuc**. Select Hosts Tab and edit the default entry there. Double Click on the item in the List. A window pop up as in Fig 6. Enter the IP Address **10.0.0.1**, Hostname **iuc** and Alias **localhost**. Give Ok. Save the Configuration. Deactivate and Activate the Device eth0. Open Terminal Window Start→System Tools→Terminal. Pass the Following Command to restart the Network Service.

```
[root@iuc root] # service xinetd restart
```

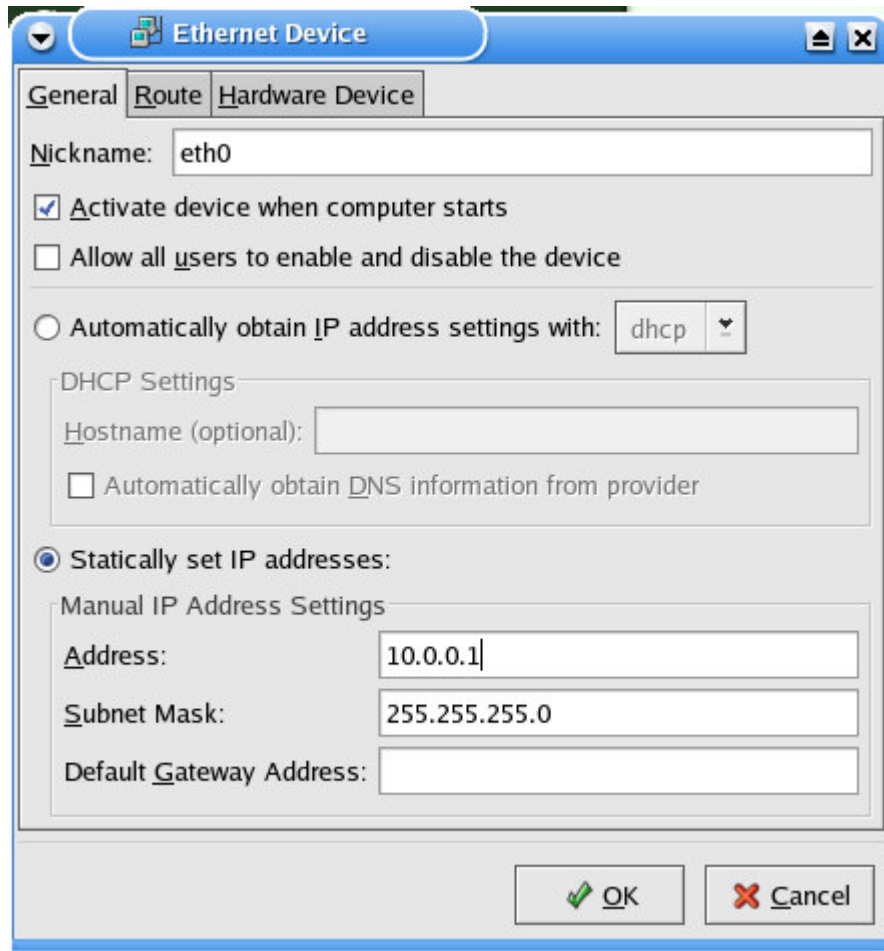


Fig 5

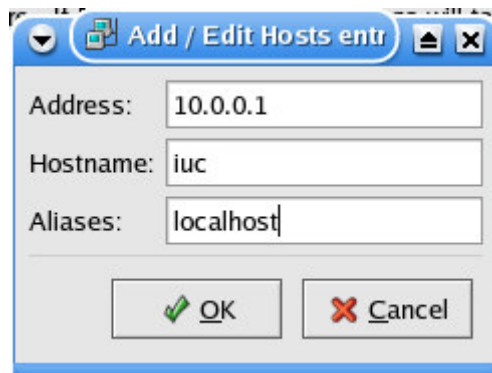


Fig 6

5.1.3 RPMs Upgrade

Linux OS Pack by default contains older versions of MySQL, Apache and PHP. Since ICUIT is using later versions, those RPMs have to be upgraded. Down load the required rpms from the net and keep them in a folder (/root/RPMs). Run the Following command at the terminal window to upgrade the rpms in order. Before starting Upgrade, Pl ensure that the list of rpms shown below are available.

The required rpms are

- httpd-2.0.40-21.i386.rpm
- MySQL-server-4.0.14-0.i386.rpm
- MySQL-client-4.0.14-0.i386.rpm
- MySQL-shared-4.0.14-0.i386.rpm
- MySQL-shared-compat-4.0.14-0.i386.rpm
- php-4.2.2-17.i386.rpm
- php-mysql-4.2.2-17.i386.rpm
- webmin-1.100-1.noarch.rpm

Commands to pass for upgrade

```
[root@iuc root] # cd RPMs
[root@iuc RPMs] # rpm -Uvh httpd-2*
[root@iuc RPMs] # rpm -Uvh MySQL-shared-4*
[root@iuc RPMs] # rpm -Uvh MySQL-shared-com*
[root@iuc RPMs] # rpm -Uvh MySQL-server-4*
[root@iuc RPMs] # rpm -Uvh MySQL-client-4*
[root@iuc RPMs] # rpm -Uvh php-4*
[root@iuc RPMs] # rpm -Uvh php-mysql*
[root@iuc RPMs] # rpm -Uvh webmin*
[root@iuc RPMs] # service mysql restart
[root@iuc RPMs] # service httpd restart
```

Check that there are no Error Messages during Upgrade. Ensure that 'mysql ended' error is not there after restart of mysql.

5.1.4 MySQL User Creation

After upgrade of the MySQL rpms, it is necessary to register the root password say for e.g if the root password is '**myiuc**' for MySQL. You have to create one mysql user in the name **iucit** with password **myiucit** for IUCIT to work. The iucit user should also be allowed to access your server from the Client machine. Let the IP address of the Client machine be '10.0.0.X'. Pass the following commands.

```
[root@iuc root] # mysqladmin -u root password myiuc
[root@iuc root] # mysql -u root -p
Password:myiuc
mysql>grant all privileges on *.* to 'iucit'@'iuc' identified by 'myiucit';
mysql>grant all privileges on *.* to 'iucit'@'10.0.0.X' identified by 'myiucit';
ctrl+c
[root@iuc root] #
```

5.1.5 IUCIT Source Copy

Replace your old **html** Directory in **/var/www** with the one in the IUCIT CD. Copy the XY.sql file to /root. Right click on the /var/www/html/profiles folder, goto permissions tab, Check all the check boxes, check the **apply changes to sub folder** check box also and give OK.

5.1.6 Importing the CODES Database

The XY Codes and Subcodes are to be imported to One database in the name CODES. To do this, Open Terminal (Start → System Tools → Terminal). Pass the following command from the folder where you keep XY.sql file supplied with the IUCIT4.1 CD.

```
[root@iuc root] # mysql -u root -p<XY.sql
Password:myiuc
```

This creates a Database in the name CODES and imports the required Code Tables in to that.

5.1.7 IUCIT Registration

Open the Konqueror Web Browser (Start → Internet → More Internet → Konqueror Web Browser) and type the address <http://iuc/install.php>. Your Screen Appears as shown in Fig 7.

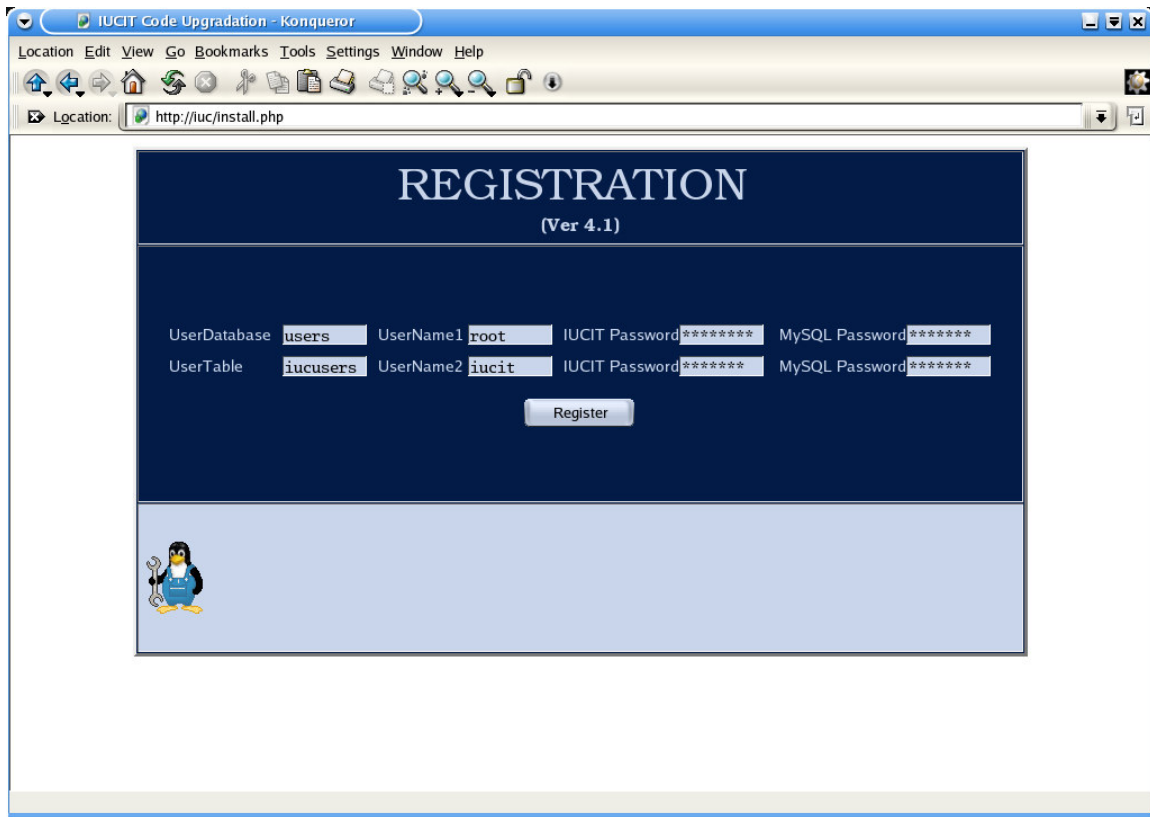


Fig 7

For User **root**, Enter IUCIT Password = **myiuc**, MySQL Password = **myiuc**
 For User **iucit**, Enter IUCIT Password = **myiucit**, MySQL Password = **myiucit**

Press Register. You should get the response as
User root Added
User iucit Added.

5.1.8 Logging into IUCIT

In the Konqueror Browser, type the address <http://iuc> and enter. You will be prompted to login by a window as shown below in Fig 8. Give the user name as **root** and password **myiuc**. A successful login invites you with Menus.

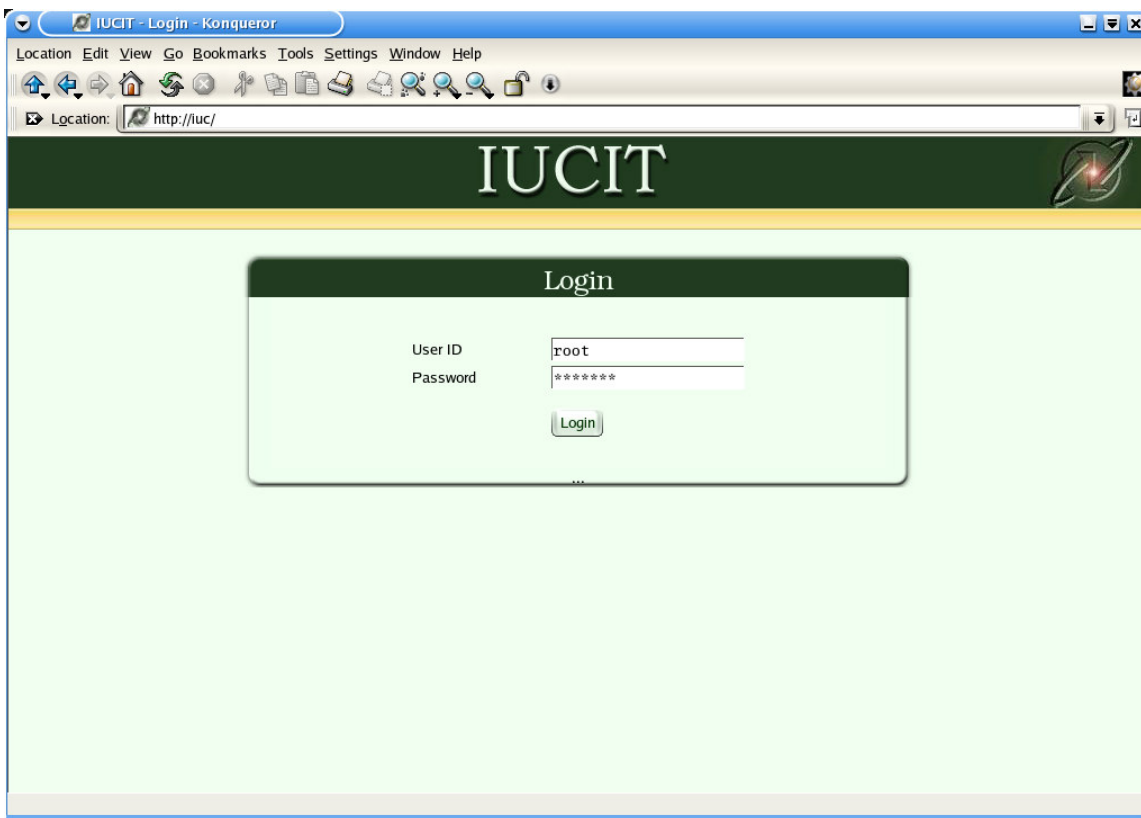


Fig 8

5.1.9 Managing Profiles

Go to Profiles → Show Profiles. It lists the available profiles. Edit the default profile with your site dependent parameters and save it in your optional name. In the list of profiles, Click on the **Activate** link to activate your Choice of Profile.

5.1.10 Creating Site Dependant Code Table

For Detailed analysis of CDRs, A site depended code table is to be created for every site. This table will contain all the available codes of India. The distance of Every Code with respect to your site is to be identified and stored in your code table which makes the code analysis process easy. Go to Tools → Code Table Creator. Select Your LDCC from the List of LDCCs and SDCC from the List of SDCCs. The Code database name, table name appear in the Code table field as entered in the profile. Select your adjacent LDCCs from the list of LDCCs holding the control key down for multi select. Press **Create Code Table** to Create your Site Dependent Code table. Go to Data → Database Explorer. Click on CODES. Click on your Code Table and verify the data (the code and the slab).

Now Your Linux Server is Ready to go with IUCIT 4.1

5.1.11 Client Configuration

Install **MySQL-Front_21.2_Setup.exe** from the IUCIT CD in the windows machine. Ping the Linux Server From Client machine and ensure connectivity. Double Click the icon placed in Desktop and register a new Connection with host as 10.0.0.1, username as '**iucit**', and password as '**myiucit**' [you should have already created the user **iucit**]. Save the Configuration. Connect to server to access the MySQL Server there. You will get connected to the Database in the server if your user id and password are valid and your host is allowed to access the server. Install the CDRX Pack from the IUCIT CD.

5.2 IUCIT on Windows

IUCIT Scripts are common for windows OS also. We have to attain the WAMP configuration on windows. Since NTFS file system supports larger data file size, it is better to load windows on NTFS.

5.2.1 Xampp Installation.

Install the XAMPP Pack available in the WIN-IUCIT folder of IUCIT CD with default options. It extracts lot of files and shows the progress message. Successful Installation of Xampp leaves the Xampp control panel shortcut in the Desktop.

5.2.2 MySQL Installation.

Xampp installs mysql 4.1 which will give compatibility errors with IUCIT. Hence the mysql 4.1 should be replaced with mysql 4.0. Extract the mysql-4.0.18-win.zip file in the WIN-IUCIT folder of IUCIT CD. Install mysql 4.0 with the setup file extracted with default options.

This installs mysql in c:\mysql folder. Replace the c:\apache\friends\xampp\mysql folder with c:\mysql folder. (Don't Overwrite. Replace).

5.2.3 Starting Services

Open the xampp control panel at the desktop. Start Apache. Start MySQL.

5.2.4 MySQL User Registration

Execute the following command to Register the MySQL Users.

```
C:\apache~1\xampp\mysql\bin> mysqladmin -u root password myiuc
C:\apache~1\xampp\mysql\bin> mysql -u root -p
Password:myiuc
mysql>grant all privileges on *.* to 'iucit'@'localhost' identified by 'myiucit';
ctrl+c
C:\apache~1\xampp\mysql\bin>
```

5.2.5 IUCIT Source Copy.

Copy the **iucit** folder from WIN-IUICIT folder of IUCIT CD and place it in the c:\apache\friends\xampp\htdocs folder.

5.2.6 Importing the CODES Database

The XY Codes and Subcodes are to be imported to One database in the name CODES. To do this, Open Terminal (Start → System Tools → Terminal). Pass the following command with full path of XY.sql (D:\WIN-IUCIT\XY.sql).

```
C:\apache~1\xampp\mysql\bin>mysql -u root -p< XY.sql
Password:myiuc
```

This creates a Database in the name CODES and imports the required Code Tables in to that.

Continue with Section 5.1.6 with address <http://localhost/iucit/install.php> for IUCIT registration and <http://localhost/iucit> for working on any Internet Browser. Continue up to Section 5.1.9.

6. IUCIT Links

Successful Login to IUCIT shows you the current version and the Links to work with. The available Utilities are categorized into Data, Reconciliation, Billing, Tools, Profiles and Help.

6.1 DATA

6.1.1 Database Explorer

This link is used to explore the Databases available in the MySQL server. The Available Databases are listed in four columns. The names displayed are Hyper Links and click on them will show the tables available inside. The list of table shows the number of records inside. Further click on the names of the table will show structure of the tables. View link will show the data available inside. Delete will drop the table. The tables with the word **user** in their name are hidden. The table names with **leading underscore** are hidden. The hidden temporary tables can be seen through **Clean Database** utility and removed.

6.1.2 Create Database

New Databases can be created using this Link. The name can be any alphanumeric string. If a database already exists it gives an error message. On Linux, the names of database and tables are case sensitive and not on windows.

6.1.3 Create Table

New Tables of [IUC Standard Structure](#) and custom structure can be created using this link. Choose the Database where you want to place the new Table. By default alphabetically first database is selected. Give any name for Table and choose the structure of the table either as BSNL or Private to create standard tables. If switch type is chosen as EWSD, in the BSNL Structure, A new field conn_id is added which is used for packing during Comparison. If custom structure is chosen then a new text box is visible to get the optional parameters. Enter values in the text box like **'name Varchar(20), userid varchar(20), age int(3)'** to create table as you like.

The small report window below shows the result. By default the table can be maximum of 2 GB on windows and 4GB on Linux. Approximately 8 Crore records can be inserted into One MASTER Table created. Further insertion will fail. To modify the tables to accommodate more records check the **BigTable** option. This makes the table to accept 20 Crores Records. The limit can be further increased by modifying the script.

6.1.4 Copy Table

The existing tables can be copied from one database to another with structure and data or structure only. One or more number of tables can be copied simultaneously. Chose the source and destination Databases in the drop down menu. The list box below the source databases shows the available tables in that database. Select one or more tables in the list box and press copy. The report is displayed in the report window below. If copy with data option is checked then the tables are copied with data else only structure is copied. Pl note that indexes in the tables are not copied. This link is useful to begin the IUC Processing for a new month. The structure of all the Master Tables in the previous folder can be copied to the new Database without data without the pain of creating all the tables once again.

6.1.5 Import Text File

Generally the Data is made ready only at the client machine since, site specific drives and the binary to text conversion programs are available in the windows machine only. In future the binary to text conversion programs are to be released to work on Linux. These programs are written in C++ and works very fast when compared to the one developed on VB. The Drives also can be mounted on Linux. At that time this link is very important to upload your data to MySQL. Choose a Database name and an existing table name and Click on the browse button to select an input text file from the local machine. Choose Correct line terminator depending upon on which OS the file is created. Press upload to export the data to MySQL. The Report window shows the Number of records uploaded.

6.1.6 Export Table Data

The data from the Tables can be dumped into text file in CSV format using this Link. This is useful to take the backup of small tables and also to dump the Error records of private operators. The CSV file will be dumped in the dump folder set in the profile. Write permission should be free for apache to handle the folder. Otherwise dumping will end with error.

6.2 Reconciliation

6.2.1 Sub Table Creator

In reconciliation process, the Master Tables should not be tampered and hence sub tables are prepared to work with. Normally BSNL Master Table contains records of all the calls processed in that exchange (if CDR enabled). It contains records of a full month. Private Master Tables contains records of that particular Operator for a full month. To compare the records, sub tables are prepared for both BSNL and Private.

6.2.1.1 Inputs

The **Database** Field shows a drop down list box showing all the available databases. The working database may be selected from the list. **Master Table Name** field also shows a drop down list box showing all the tables from the database chosen. The Master table from which a sub table is to be created is to be selected. **Sub Table name** accepts any string. Choose a name of meaningful words to understand about the sub table (eg. ac_may04, bac_may04, bsnl_ac_may04 etc). Enter the period of selection. Change the Month field of Starting Date option. The period, that is starting date, end date, end month are set automatically. The year field shows the current year. There are other four options **calling Number, Called Number, Incoming TGP ID and Out Going TGP ID**. None of these four is mandatory. These four items accept values in comma separated format. The final one is **Prefix Area Code Field** where the default area code is displayed and can be changed.

6.2.1.2 Background Operations

A sub table of specified name is created with fields depending on the structure of Master Table without Data. If sub table already exists, that is not dropped automatically but error message will be displayed. The sub table is altered and four more fields (**extdur, id, co_id and flag**) are added. The **id** field is set as Auto increment and Primary Key. The data of specified period matching other conditions is selected from the Master Table and inserted into the sub table in the order of date and time. Every Record in the sub table is given a unique number. The Leading '95' from **called_id** and leading '0' or '91' from **calling_id** are trimmed. If Area code is given, then area code is prefixed in the **called_id** wherever the **called_id** is not beginning with the given area code. Missing of Area Code in the Sub table will lead to Synchronization Failure during comparison.

6.2.2 Table Comparer

The comparer does the major part of the work in reconciliation. Every record of Private Operator is compared with the BSNL Database and valid records are marked. The number of records in BSNL Database is immaterial. But every record of Private Operator should exist in the BSNL Database. The valid and invalid calls declared finally by IUCIT are explained schematically in Fig 9

6.2.2.1 Inputs

The fields, **Database, BSNL Sub Table, Private Sub Table** fields shows drop down list boxes with available databases and tables. The working Database, private and bsnl sub tables should be properly selected. Choose the right switch type (BSNL Switch) in the Switch Combo. The ISD TGP field accepts any valid incoming TGP Name. If both BSNL and Private Switches are EWSD, select AXE Switch.

6.2.2.2 Background Operations

During comparison, the Private Sub Table is altered and two new more fields **newdate** and **newtime** are added. Before starting comparison, the time of the records of Private Operators is to be synchronized with the BSNL Records. After synchronization, every record of Private Operator is given a new date and new time. The comparison process is done in many levels to match maximum possible records and hence to reduce the burden of the reconciliation officer. [Different flags](#) are marked in both the BSNL and Private tables, which are permanent even after the

comparison process is completed. The stages of comparison differ from switch to switch. The levels performed by comparer depend on the values entered in the comparer form and the values set in the configuration File. The different stages of comparison are explained below. Before starting the stages, to initialize the tables, the time and date of each record is copied to new time and new date. The flag and co_id are set to '0'. This Initialization is done every time the table is put for comparison

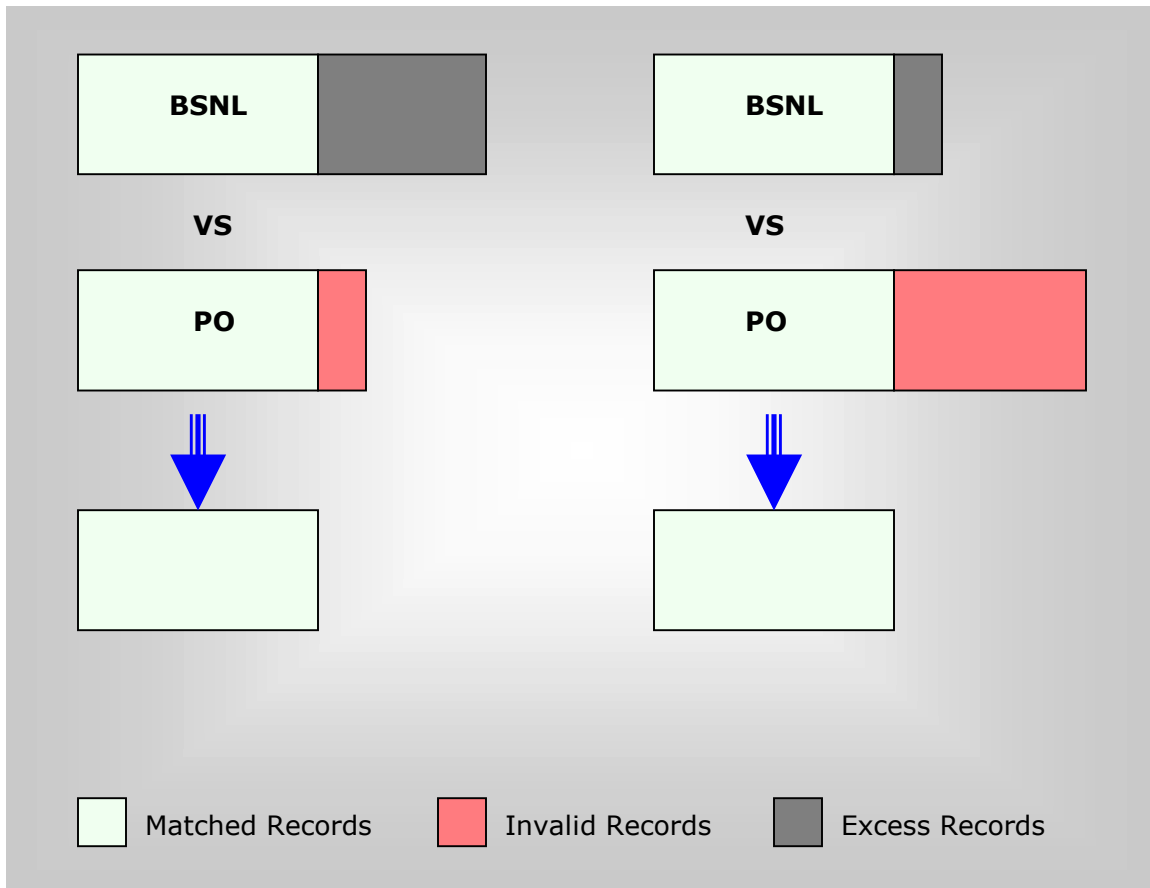


Fig 9

6.2.2.2.1 Data Packing

This level is performed if the switch type is EWSD. Unlike other Switches, EWSD breaks a long duration record for every half an hour by default and writes it as different records. That is why additionally the conn_id field is added for EWSD. Every matured call is given a separate connection ID in EWSD and it is same even if the record is broken in to many. Since other switches write only one record at the end of the call, it is necessary to merge the broken records of EWSD into single record for the comparer to match them. This merging of broken records is done in the packing stage. Records with same **conn_id** are merged into one with time as the time of first record and duration as the sum of durations of all the records. Because of merging, the number of records in the sub table may be reduced after comparison.

6.2.2.2.2 Time Adjustment

This Stage is specific to OCB Switch. OCB Records End time of a call by default, while other switches record Origination time of the call in the CDR. A call of 2 hours duration and originated at 23:00 Hrs today will be recorded with time as 01:00 Hrs of next day by OCB. This discrepancy in time of records is adjusted in this stage. This is achieved by adding the duration of the call with the origination time of the Private Records. In this adjustment even the date of the record may change. Hence a New Date and New Time is given to each record and they are entered in the respective fields. Note that the Private Records only are given new date and time with respect to BSNL Records.

6.2.2.2.3 Time Synchronization

This stage is common to all switches. It is practically too difficult to keep the time of all the exchanges same. BSNL Records of a particular exchange are based on the clock of that exchange. Similarly PO Records are based on their Clock. These two clocks are normally not in phase. Hence the PO Clock is to be synchronized with the clock of BSNL. This is simply achieved by synchronizing the records. IUCIT Creates two temporary tables one for BSNL with sample records having time like 10:xx and another table for Private Operator with sample records having time like 10:0x. That is Daily one hour (10:00 to 10:59) records are selected from BSNL and 10 Minutes (10:00 to 10:09) records are selected from PO Table. Each record in the PO temporary table is compared for calling_id and called_id with the records in the BSNL temporary table and the time difference of the matched records is noted. The best value of time difference is chosen for every day. The new time of all the records of a particular day are added with the time difference observed for that day. Thus after synchronization the time of Private Operator comes to a tolerance of one second with BSNL. The sample taking time can be modified in the Configuration File. You will get a Synchronization Failed Message only if none of the PO record matches with BSNL.

6.2.2.2.4 Strict Call Matching

After synchronization the tables are ready for comparison. In Strict Comparison Every record of PO is checked for a match in the BSNL table. The date, time, calling_id, called_id and duration are checked during matching. A specified tolerance in time of the calls and in duration of calls is allowed during comparison. In every stage of matching, a time tolerance of ≤ 2 seconds and duration tolerance of ≤ 2 seconds is working by default. These values can be changed in the Configuration File by setting values of \$defTime and \$defDur. The actual limit is less than the value marked in the Configuration File. Every matched record is marked with the partner ID in the co_id field. The id of BSNL table is co_id of Private Table and vice versa. The Strictly Matched calls are marked with '1' Flag, also the calls with zero duration are marked with 'Z' flag. The Following example shows '1' flag marking by IUCIT.

BSNL

Date	time	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522631608	9842250005	1299	0	657	874	1

Private

Newdate	newtime	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:45	4522631608	9842250005	1299	0	874	657	1

6.2.2.2.5 Redirected Call Matching

A call pumped from BSNL to Private is diverted by private to another number is a diverted call in IUCIT. In this case all the fields of the records will match except the called_id field. The left out calls with flag '0' from the PO table are copied in a temporary table (both BSNL and PO), and compared again relaxing called_id. The matched calls are marked with 'R' flag and co_id is noted in both tables. The records marked with 'R' flag are updated to Sub Table and deleted from the temporary table. An example for redirected call is given below.

BSNL

Date	time	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522631608	9842250005	1299	0	657	874	R

Private

Newdate	newtime	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:45	4522631608	9842223344	1299	0	874	657	R

6.2.2.2.6 Split/Slip Call Matching

The record split is normally observed only with CMSP. The Cellular billing pattern is different and there are many chances for splitting of one record into many. The left out calls in the temporary table are checked for split. That is, records with same calling_id, called_id and date are grouped in to one record with cumulative duration. This record is compared with BSNL Record. The id of BSNL is entered as co_id for all the records in the group. If record splits into two different days then IUCIT will not match it. The Split matched calls are marked with 'S' flag. Not only for the CMSPs, if the private operator uses EWSD switch then also for every 1800 seconds his record breaks. Mostly 'S' flag is marked for split calls. That is, for a call from BSNL table, more than one calls will be there in the private table. Some times, if there is some slip in the time of private operators in the middle of the day by few seconds, the calls will be matched in the strict matching. But, those single slipped calls differ only in time and these calls are marked with 'S' flag and accepted. For these calls only one partner will be available in the private table. The following example explains the split records.

BSNL

Date	time	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522631608	9842250005	1299	0	657	874	S

Private

Newdate	newtime	calling_id	called_id	duration	Extdur	id	co_id	flag
2004-05-02	04:52:45	4522631608	9842250005	600	0	874	657	S
2004-05-02	05:02:45	4522631608	9842250005	135	0	2457	657	S
2004-05-02	05:05:00	4522631608	9842250005	564	0	3650	657	S

6.2.2.2.7 Higher Duration Call Matching

The tolerance in duration is set by default as 2 Seconds in the configuration file. That means that the difference between duration of BSNL and PO records should be less than or equal to 2 seconds. If the records match except duration, and if the private operators duration is higher than BSNL duration then the records are marked with 'H' flag and the difference in duration is entered in the **extdur** field. 'H' flag is marked

on both tables and the extdur value is entered only in the private operator's table. While preparing report, the extra duration marked will be reduced from the actual duration by the reporter. The following example shows higher duration call matching.

BSNL

Date	time	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522631608	9842250005	1299	0	657	874	H

Private

Newdate	newtime	calling_id	called_id	duration	Extdur	id	co_id	flag
2004-05-02	04:52:45	4522631608	9842250005	1315	16	874	657	H

6.2.2.2.8 Lower Duration Call Matching

If the duration claimed by private operator is less than recorded by BSNL then those calls are marked with 'L' flag and accepted. The extra duration (the difference) is entered in the **extdur** field of BSNL sub table and not in private sub table. This is just for information.

BSNL

Date	Time	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522631608	9842250005	1299	300	657	874	L

Private

Newdate	newtime	calling_id	called_id	duration	Extdur	id	co_id	flag
2004-05-02	04:52:45	4522631608	9842250005	999	0	874	657	L

6.2.2.2.9 PBX Call Matching

In some EWSD Exchanges, it is observed that when a call goes from a PBX to Private network, the group number (Prime Number) is recorded in the BSNL CDR where as the Original Number is sent to the Private Network. In these case, the first 4 or 5 digits of the calling_id in both the tables match but not for all digits. These calls are valid and marked with P flag.

BSNL

Date	Time	Calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522323500	9842250005	125	0	657	874	P

Private

Newdate	newtime	calling_id	called_id	duration	Extdur	id	co_id	flag
2004-05-02	04:52:45	4522323512	9842250005	125	0	874	657	P

6.2.2.2.10 Optional Rejection

This stage reduces the work of reconciling officer in the verification stage. Those calls, which are not relevant to our database can be made automatically rejected by ICUIT by setting the flag \$defMiss to 1 in the Configuration file. This flag is normally set. The temporary table data is updated to sub table after every stage and finally the unmatched calls only are there in it. Now the availability of any unmatched

record in the BNSL with the same calling_id and called_id is checked for every record in Private. If available, then the record is left with 'O' flag else rejected with 'X' flag and updated to sub table. There is one subset of this stage which rejects only those irrelevant calls with duration less than 3 seconds. That is enabled by setting \$defL3R flag to 1. Note that \$defMiss supercedes \$defL3R. Finally all the valid calls without CLI are marked with flag 'C' meaning without CLI. These calls are separately placed in the report.

6.2.2.2.11 Duplicate Identification

It is observed in the Private Tables, that there are many duplicate records and even triplicate records. That is, similar records or are similar record but one second difference in time or similar record but one second difference in duration. If more number of similar records are there, strict comparison allots same co_id for more than one record in the PO Table. If co_id is not unique then, the first record of many is accepted with '1' flag and the remaining are marked with 'D' flag meaning Duplicate. The following example shows how duplicate is identified. For convenience only selected fields of BSNL and private are shown.

BSNL

date	Time	calling_id	called_id	duration	extdur	id	co_id	flag
2004-05-02	04:52:46	4522631608	9842250005	1299		657	874	1

Private

newdate	newtime	calling_id	called_id	duration	Extdur	id	co_id	flag
2004-05-02	04:52:45	4522631608	9842250005	1299	0	874	657	1
2004-05-02	04:52:45	4522631608	9842250005	1299	0	875	657	D

6.2.2.2.11 ISD Calls Marking

Since ISD calls are very important by tariff, they have to be separately identified and marked as ISD Calls. ISD calls can not be simply marked with respect to the CLI it brings. ICUIT already trims all the leading zeros in the sub table. All those calls which have come through ISD TGP from Level I TAX to Level II TAX or from Level II TAX to Local Exchange are marked as ISD calls irrespective of the CLI it brings. So it is necessary to have a separate Trunk Group for incoming ISD calls in all the exchanges and it is a standard instruction too. Normally only one such TGP will be there and that is given as input in the comparer. Note that calling_id is not verified for the calls coming from ISD TGP, since the length of CLI may vary and will not fit into limits specified by some switches. If ISD Tgp value is not entered, then this stage will be skipped and ISD calls will not be marked. It is necessary to enter this parameter for BSOs. Though it is not important for CMSPs by tariff, calls with wrong CLI or partial CLI through this tgp also will be matched by IUCIT.

6.2.2.3 Valid and Invalid Calls

The comparer marks different flags during comparison. There are about eleven different flags in IUCIT comparison. The flags are grouped and classified as valid and invalid. The intermediate reporter after comparison reports in this classification. There is one more flag which comes only after verification is flag 'A'. There are some records which cannot be decided by IUCIT and comes in the verification stage for

manual verification. Manually accepted calls are marked with flag 'A' meaning Accept. The following list shows the valid and invalid calls in IUCIT.

Valid Calls

- Strictly matched calls [1]
- Redirected Calls [R]
- Split Calls [S]
- Zero Duration Calls [Z]
- Higher Duration Calls [H]
- Lower Duration Calls [L]
- ISD Calls [I]
- CLI less Calls [C]
- Accepted Calls [A]
- PBX Calls [P]

Invalid Calls

- Duplicate Calls [D]
- Rejected Calls [X]
- Not Verified Calls [0]

6.2.2.4 Errors during Comparison

There are two important errors that may occur during comparison and strongly affect the progress. When any of this error encountered, Comparer stops with an error message. **Synchronization Failure** is one of the two. This error occurs when wrong tables are given for comparison. This means that none of the record is matched in any day during sample check. The second one is **Large number of Records didn't match** error. During strict comparison at least 50% of the total calls in the PO Sub Table should be matched, else this error message is displayed and comparer stops, since further proceeding will lead to huge delay in comparison.

6.2.2.5 Intermediate Report

The live progress of the comparer is displayed in the browser and ends with an intermediate report. The report summarizes the result and informs the valid and invalid call. Summary is displayed in Yellow color. The valid call detail is displayed in green and the invalids in Red. The number of not verified calls displayed comes in the verification stage.

6.2.3 Verification

There are some records, which could not be decided by IUCIT [calls with flag '0'] and comes to verification stage for manual verification. All these calls are to be either accepted or rejected to avoid an error message in the Reporter. Normally the number of records to be verified is very less and in two digits only for New Tech Exchanges. The Verification Window is split into three sub windows. The top one shows a maximum of 100 Records to be verified at time. The bottom left window displays records from BSNL temporary table and bottom right window displays records from Private temporary table. The windows show all the unmatched records with same calling_id and called_id.

6.2.3.1 Inputs

The dropdown fields, **Database, BSNL Sub Table, Private Sub Table** accepts current working Database name, BSNL Sub table for the Particular Private Operator

and Private sub table name respectively. **Starting Record Number** is by default 0 and can be changed based on the requirement.

6.2.3.2 Working on Temporary Tables

For Quick reference of BSNL and Private Records, the records with flag '0' are selected from both BSNL and Private sub tables and inserted in to temporary tables. This is done when **Execute** button is pressed and hence a small delay in displaying records can be observed. Now the Execute button is replaced with **Update** and **Discard** Buttons. The first 100 records to be verified are shown followed by an **Enter** button at the end of page. Every record has a **BSNL** and **Private** buttons with three radio buttons **Keep**, **Accept** and **Reject**. Pressing the BSNL and Private buttons shows unmatched records from respective temporary tables in respective windows. The records can be manually verified and decided to either Accept or reject and the radio button is set in the corresponding position. The records to be decided later may be kept at Keep Position. When all the records or some records are decided, the enter button at the bottom of the page is pressed to enter the settings in the temporary table. Now the remaining records with '0' flag are shown in the verification window. This operation is repeated to decide the validity of all the records. Finally the entries in the temporary are updated to Sub Table by pressing **Update** button. If any error occurred during verification, the temporary tables may be reset by **Discard** button.

6.2.4 Super Check

This link is for higher officials to do sample checking after reconciliation. The records rejected by IUCIT can be analyzed and accepted can be verified easily in this stage. In this stage no modification can be done to the records. It works directly from the sub tables and hence depending up on the volume of records, the stage may delay to show records. Like verification stage, the window is divided in to three for easy checking.

6.2.4.1 Inputs

The dropdown fields, **Database**, **BSNL Sub Table**, **Private Sub Table** accepts current working Database name, BSNL Sub table for the Particular Private Operator and Private sub table name respectively. **Starting Record Number** is by default 0 and can be changed. There is another dropdown list box showing all the available flags of IUCIT with the **show** submit button. Select required flag and press **show** to get the records displayed.

6.2.4.2 Working

Like verification stage, every record displayed is having two buttons **BSNL** and **Private**. Pressing these buttons shows records having the same id and co_id from both tables. Since **id** is primary key the display will be quick for one to one matched calls. For flags like 'S', 'D' and 'X' where the matching is not point to point the display may be slightly delayed depending upon the volume of the records.

6.2.5 Code Analyzer

This stage is necessary for settling bills of BSO. For CMSP there is no distance based billing. But for BSO the distance of origination of the call is to be taken into account, since the tariff differs on distance. Also till Feb 2004, the distance only was deciding

the tariff. But after Feb 2004, the nature of the originating call is also a deciding factor. ICUIT 4.1 automatically identifies the Phase of Reconciliation and do code analysis accordingly. Code Analysis need not be performed for CMTS since flat rate tariff. For the data after February 2005 code analysis need not be performed for any type of Operator. If given for code analysis, the analyser quits with intimation.

6.2.5.1 Inputs

The dropdown fields **Database**, **Private Sub Table** accepts current working Database name, and Private sub table name respectively. The field **BSO Fixed levels** accepts the fixed levels of BSO with Area code in comma-separated format. **Code Table** accepts a valid code table name in **Database.Table** Format. The area code field is necessary to mark the local calls separately in the tandem exchanges.

6.2.5.2 Background Operations

Since there are four different phases of ICU, four different methods of code analysis are played. The Site Dependent code table created initially supports all different phases. The Entries in the Subcode may vary in different phases. Care must be taken to update the subcode entry for different phases.

6.2.5.2.1 Phase I (Before Feb 2004)

Three more fields, **cat**, **lcat** and **slab** are added to PO Sub Table. The **cat** field is to enter the type of the originating number, **lcat** field is to enter the type of terminating number and **slab** field is to enter the distance of the call. Before starting code analysis, the sub table is initialized by setting the cat field of the records as '0', lcat field of the records as 'M' and slab field as NULL. Analysis runs in five levels. In the first level, the codes in the code table with length 6 are matched with the calling id in the sub table and the slab, cat fields of the code are entered in the sub table. Level six compares the two digit codes in the code table with the Sub Table. The marked records are not remarked in the analysis. Some records may fail in code analysis and they can be found out by checking the cat field. The PO fixed levels entered is compared with the called_id of each record and the lcat field is set as 'F' if matched. Any record having cat value '0' after code analysis is skipped from code analysis. The reason for skip should be analyzed and missing codes may be entered in the code table. The different possible values of **cat**, **lcat**, **slab** are given in Appendix.

6.2.5.2.2 Phase II (After Feb 2004)

Four more fields, **cat**, **lcat**, **slab** and **len** are added to PO Sub Table. **cat** field is to enter the type of the originating number, **lcat** field is to enter the type of terminating number, **slab** field is to enter the distance of the call and the **len** field is to mark the length of the Area Code in the calling_id. Before starting code analysis, the sub table is initialized by setting the cat field of the records as '0', lcat field of the records as 'M', slab field as NULL and len field as '0'. The analysis is done in two parts. First part runs for 5 times with longest code first. The analysis is similar to Phase I, but the length of the area code is limited to four digits and hence only four levels instead of six. There is one sub table named subcode in the code database which has standard sub levels. The second part of the analysis starts with subcode. Two digits after Area Code in every record is analyzed and the cat of the sub code is concated with the cat marked during First Part analysis. For example, a record with calling_id 4522632608 is found as Intra Circle call (IC) by Code Analyzer in Part One analysis by matching the Area code 452 (Madurai). Then in part two the next two digits '26' is to be analyzed. The subcode table has an entry '2' with cat 'BF' (BSNL Fixed).

Hence the cat of the record will be marked as IC + BF = ICBF. The PO fixed levels entered is compared with the called_id of each record and the '**lcat**' field is set as 'F' if matched. The Structure of code table and subcode table are shown in Appendix. No need to do code analysis after Feb 2005

6.2.6 IUCIT Reporter

IUCIT Reporter gives the result in a printable format with three different types. All the records should be verified before report preparation to avoid a warning in the report. Reports for CMSP can be prepared easily and immediately after comparison and verification, since the payment is on flat rate and there is no need for code analysis. The reporter will give only the valid **MOUs** that can be paid. There is also provision to get the result in Rupees, for that you to enable proper bits in the configuration file.

6.2.6.1 Inputs

The **Database** and **Sub Table** fields accept the name of the working database name and Private Operator Sub Table name respectively. **Name of PO, Invoice date,** and **Invoice number** are other input fields. The **Type of Operator** is an important parameter, which decides the Report Style.

6.2.6.2 Types of Reports

There are three different types of reports available. The **Summary Report** gives information about the total calls Offered, valid, invalid calls and their sub divisions. **The Detailed Report** gives detailed information about the valid calls. Note that the Detailed Report informs nothing about the invalid calls. There is no difference in the first two report types in Phase II. But Detailed Report gives different style report in different phase for different operator types. For Phase III and IV, The report simply gives the Total MOU date wise since flat rates applied.

6.3 Billing

The following links are useful to generate bill towards private operators. Billing is easier when compared with the reconciliation since there is no need for comparison. All the records recorded by BSNL are genuine records and all are accounted in raising the bill.

6.3.1 Billing Table Creator

To raise every bill, it is necessary to create a separate billing table towards every operator. Any future query about the bill can be cleared using this table instead of browsing the Huge Master table. It is easy to give the CDR for the bill raised towards any private operator (if permitted)

6.3.1.1 Inputs

The working **Database** name and **Master Table** name must be selected from the drop down list boxes. The **Sub Table** name accepts any string. Choose a name of meaningful words to understand about the billing table (eg. Bill_acmay04, Bac0504, Bill_Aircel_may04 etc). Enter the period of selection. Change the Month field of

Starting Date option. The full month period that is, the starting date, end date, end month are set automatically. The year field also shows the current year. Set the starting year, the end year is set automatically. There are four other options **Calling Number, Called Number, Incoming TGP ID and Outgoing TGP ID**. None of these four is mandatory and accepts more than one value in comma separated format. **And** operator works between these four fields and hence care should be taken. The **Prefix Area Code** Field shows the default area code from the configuration file. This field is filled up, if prefixing of area code is necessary in the called_id field else cleared. The last one is prefixing '0' in the called_id. If the switch of PO is EWSD then it may be programmed not to send the leading '0' in the called_id. In such cases the leading zero may be prefixed. The VSNL IC option is to be checked if the Table is prepared for VSNL.

6.3.1.2 Background Operations

A sub table of specified name is created with fields depending on the structure of Master Table without Data. If sub table already exists, that is not dropped automatically but error message will be displayed. The sub table is altered and five more fields (**cat,slab,len,lc,ilen**) are added. The data of specified period matching other conditions is selected from the Master Table and inserted into the sub table in the order of date and time. If Area code is given, then area code is prefixed in the **called_id** wherever the **called_id** is not beginning with the given area code. This case is possible only at tandem exchanges. A '0' is prefixed in the called_id if the appropriate check box is checked. If VSNL IC option is checked, then home country calls also are included.

6.3.2 Code Analyzer

Bill is raised from CDR depending upon the distance up to which the call is carried by BSNL and also the nature of the calling line. Hence the distance carried must be calculated from called number of the call and the nature of the calling line (that is either fixed line or mobile line) must be identified from the calling_id. Here Code analysis plays an important role.

6.3.2.1 Inputs

The working **Database** name and **Master Table** name must be selected from the drop down list boxes. **Code Table** accepts a valid code table name in **Database.Table** Format. The area code field is necessary to mark the local calls separately in the tandem exchanges. The **ISD TGP** name may be entered for BSO. The POI Type field is set properly to reduce the work of Code Analyser in some special cases. The roaming subscriber call trunk groups should be given in the order of National Roaming STD, National Roaming ISD, International STD, International Roaming ISD separated by comma. If these trunk groups are not given roaming subscribers are identified using CLI in IUC Phase IV and not for IUC Phase III. The allowed fixed levels of LPOs can be given to restrict only the Allowed levels. Other calls will be marked as Wrong CLI.

6.3.2.2 Background Operations

The called number is analysed by CA and the distance of the terminating point from the point of hand over is identified and marked in the slab field. Also the category of the call (IC,OC,ICBC,OCPM) is identified and entered in the cat field. The length of the Area code is also entered in the len field. The called_id is further analysed in depth after the area code and the sub level is found out and sub category (eg BF, PM) is concatenated with the category. Code analysis ends with this if the POI type is CMTS or ILDO since the origination call type is immaterial. In case of BSO or NLDO the calling_id also is analyzed in depth and the process is repeated. Finally, the category of the calls with no CLI are marked with 'NCLI' and calls with wrong CLI are marked with 'WCLI'. If ISD TGP name is entered, all the calls through that tgp are marked with cat as 'I'. The roaming subscriber records will be marked with cat NRS, NRI,IRS and IRI flags. If allowed levels are given, other calls will be marked with cat WCLI.

6.3.3 Billing Reporter

Billing Reporter gives the result in a printable format. The output is normally in MOU only. By setting Suitable options in the profile, the MCUs and Amount can be shown in the Report. Billing Reporter automatically identifies the period of the file and prepares report accordingly. It is suitable for all 4 IUC Phases.

6.3.3.1 Inputs

The working **Database** name and **Billing Sub Table** name must be selected from the drop down list boxes. **Name of PO, POI Types** are the other parameters to enter. Type of POI decides the Report Style.

6.3.3.2 Background Operations

A temporary table is created with the number of calls, total duration and grouped by cat, lcat and slab. From this table the report is generated as per Regulation Instruction. Depending upon the phase of IUC, different files (bph2.php, bph3.php, bph4.php) are used internally. The Reporter Finally tallies the Total in the report with the actual total. In case of difference, it intimates.

6.4 Profiles

The different variables used during Reconciliation and Billing are stored in profiles. This concept is useful for centralized processing units. Different profiles can be created for different sites and activated when required.

6.4.1 Show Profiles

This lists the available profiles. The profiles are stored in a folder named **profiles** inside the script folder in text format. Apache should have write permission to write into that folder for profile modifications. Activation and modification of profiles will not be possible if there is no write permission for Apache. You edit or delete profile using the links shown.

6.4.2 New Profile

Fresh profile can be created using this link.

6.5 Tools

Some important tools to assist the process of reconciliation and billing are grouped in this section.

6.5.1 Data Integrity Analyser

Before starting either Reconciliation or Billing, it is a must to check the Integrity of the Data Uploaded by BSNL in the Master Table. There should not be any gap in data. Bulk data loss can be easily identified by simply sorting the records date wise. But small gaps of 5 minute or 10 minute gaps may not be possible by normal sorting. This tool analyses a master table and gives a graph of daily record count for easy perception. Then it analyses the record deeply through every minute and gives out a report of missing periods. A high accuracy of 1 minute gap can even be identified by this Analyser. The graph plotted shows Sundays in different color for clarity. A click on any date displayed will show you the graphs of hourly traffic. One can start processing of Bills only after getting a satisfactory report from the Data Integrity Analyser.

6.5.2 Code Table Creator

Creation of Code table for a specific site is one time job which is done initially. The method of distance calculation is different for BSO and NLDO. Also in some circles like Tamil Nadu there are two circles called Chennai Cellular Service Area with SDCAs (044,041xx) and the rest of Tamil Nadu as Tamil Nadu Circle. This type of consideration is done only for Cellular and UASP calls. This Code Table Creator creates a single site dependant code table suitable for all POIs.

6.5.2.1 Inputs

The Database in which the XY matrix available is to be chosen in the Database field. Then the XY table is chosen in the table name. Give some valid name for Code table. From the Drop down list box choose your LDCC. In the SDCC box, the SDCCs related to the chosen LDCCs are listed. Choose proper SDCC. The Adjacent LDCCs list box lists all the available LDCCs of the country. Choose the adjacent LDCCs by multi selecting in the list box.

6.5.2.2 Background Operations

A code table with fields (code, cat, len, slab) is created. The own sdcc's XY co-ordinates are related with XY of every code in the XY and the distance is found out considering the POI type Adjacent LDCA concept. Appropriate value is entered in the slab field of the code table. The value of code, cat are taken from XY and inserted in to the table. The length of the code is found and updated in the len field. The code field is set as primary key. The ISD Codes of 3 Zone, 10 Zone, 14 Zone classifications if different phases are marked in the same code table.

6.5.3 Distance Finder

It is a simple tool to find the radial distance between two SDCCs.

6.5.4 Update XY

The XY table may have to be upgraded with the latest release of new codes. The field units may insert the codes into their XY and recreate the code table or they can straightly insert the new codes into their code table. The updated csv file of the XY codes is circulated through yahoo groups which can be given through this utility to upgrade the XY. This operation renames the old XY table as XY_old and imports the new into XY. After upgrade you have to recreate the code table.

6.5.5 Disk Free Space

The free available in the Disk for the data can be verified through this link. This checks the mysql data partition set in the Profile for free space and gives a graphical output. The data should be backed up and freed if the free space is less than 10%. If mysql data is kept in separate partition give the mount location in the profile. If default location (/var/lib/mysql) then give '/' in the profile. For windows version also give '/' in the profile.

6.5.6 Sort Table Data

The records, duration, units in a table can be easily sorted using this link. The output will be for the whole data in the table by default. To get only for a particular month or with any condition, give the condition (eg. date like '2006-03%') in the condition text box.

6.5.7 Clean Databases

During reconciliation and billing processes, many temporary table are created and deleted automatically by IUCIT. If the processes are interrupted, there are chances for the temporary tables left undeleted. Since these table names start with leading '_'. These are not visible in the Database Explorer. This garbage is removed by this utility. This shows the available temporary tables and clears them when ordered.

6.5.8 Add New Code

Newly introduced codes can be easily inserted into the XY or the site dependant code table using this tool. Give the New code and a reference code. The new code is added with the parameters of the reference code if available.

7. Operating Procedure

This Section explains how to work in IUCIT Step by Step. There are some work to be performed in the Client Machine, some with Server and rest with both. The total work can be divided into three stages. They are Data Conversion, Data Upload and Reconciliation. The Data for Reconciliation is made ready at the client machine and uploaded to server through MySQL Client. The uploaded data can be processed either from Client or Server. Server is preferred for reconciliation since the clarity is more in the Konqueror Browser and IUCIT is developed for that browser. Internet Explorer or Netscape also can be used from the client. The following are the steps to be followed in IUCIT Reconciliation.

7.1 Data Preparation – Work on Client Machine

The Driver software and the conversion programs are available in windows versions only at present. Hence the need for a client machine on windows. This is normally the Billing PC supplied for OCB, an X-25 Link for EWSD and any PC for Other Switches. The main work on the client is to make the BSNL and PO Data Ready.

7.1.1 Data copy to Client

Create a Folder in the name of month of processing (eg **May04**). Create three folders **Bin**, **Text** and **PO** inside it. Collect all the storage media for that particular month. The media is Cartridge for OCB, Optical Disk for EWSD, Zip Drive for AXE, Audio Tape for 5ESS and Cartridge for CDOT. The media are mounted in the drive one by one and all the data files of that month is copied into the Bin Folder. For OCB and EWSD Data file analyzers are available and hence the period of data can be found. Name the binary files in such a way that it represents the period of data in that file. NovaXchange is the software used to copy data from Cartridge to PC. Modena is used to Copy data from Optical Disk.

7.1.2 Binary to Text Conversion of BSNL Data

There are programs available for Binary to text conversion for all the switches. Alki1.4 for OCB, Simi1.7 for EWSD, Dotcy1.0 for CDOT and Rixy1.0 for AXE are the Binary to text conversion programs except Rixy, which is a text formatter. These programs can be down loaded as installable packages from <http://geocities.com/umapst/dl.html>. All the programs are developed on VB. Using these programs convert all the binary files to text files and place them in the text folder. The output options should be properly chosen to get the data in CSV [Comma Separated Value] format in a single file.

7.1.3 PO Data Formatting

The Private Operator Data is available on CDs. The format of all the private operators is changing dynamically. What ever may be the number of fields given by private operators, we need only the [IUCIT Standard Fields](#). Also the PO data is in different files date wise. From PO CD, the required fields only should be picked up and placed in a single file in CSV format. Hence a small program may be written to extract the data in required format or look in the web for such programs. The necessary Exe files are circulated in the **iucit yahoo groups**. Extract the data and place the output file in the PO folder.

7.2 Data Upload to Server

A MySQL Client tool **MySQL Front 2.2** is available in the client machine to get connected to the MySQL Server in the Linux Machine. Establish connection with MySQL. Create a new Database through tools→Create Database link. Create [IUCIT Standard tables](#) in the database for BSNL and Private Operators. Normally the Master Tables are named in capital letters. The structure of tables can be copied from one database to another through Im-/Export → Export Tables Link. It is easy to create standard tables from IUCIT Data Manager. Through the link Im-/Export → Import Text Files, import the data into tables with options **Fields Terminated** by to **'** replacing **'\t'** and **Lines Terminated** by to **'\r'** replacing **'\n'**. The import window

disappears once importing is over. Check the number of records imported and the data. All the text files for that month are imported in the BSNL table one by one.

7.3 Work on server - Reconciliation

After uploading the data to server from client, the work at Server starts. The reconciliation and billing process take time depending upon the volume of records. The typical time for comparison is one minute for one Lakh calls. Time may slightly vary with volume.

7.3.1 Data Integrity Check

Before starting any work with the Data, the integrity of the Data should be checked. Use Data Integrity Analyser in the Tools Section to Analyse the Integrity and ensure that there is no Gap in the Data. If any gap reported, then identify the missing files and add them. If any abnormal increase of records in particular days in the bar chart, then data may be duplicated. Check and Remove the duplicate files.

7.3.2 Sub Table Creation for BSNL

The private operator records are to be separated from BSNL Master Table into a sub table. Give the Database name and Master Table name as BSNL. Choose a meaningful name in lower case for Sub Table (eg bsnl_aircell_may_2004, bac_may04). In the beginning date combos, enter the month. The period is automatically set for the full month. In the called_id field enter the levels of the PO comma separated. Care must be taken to include all the levels for that operator. The Area code Prefix field shows the exchange area code by default. Remove the value if no prefixing is needed. Press Create to create a sub table for BSNL as required. Check the status window for positive report. If private data is not available, and no need for reconciliation, then skip the comparison process and go to code analyser.

7.3.3 Sub Table Creation for Private

The sub table is created similar to BSNL but without filtering any record. That is don't give any value in the called_id field. The number of records in the master table and sub table must be equal.

7.3.4 Pre Check Before Comparison

Check the data in Both BSNL and Private sub tables. The number of records in both the tables must be approximately equal. If the variation is more than 20% check again for missing Data.

7.3.5 Table Comparison

Fill up the required data in the Table Comparer and Start Comparison. Since different treatment is given for different switches, more care should be taken in selecting the switch. Enter the ISD TGP to mark the ISD calls separately.

Remember, only those calls coming in that trunk group are marked as ISD and not those with leading '00' since there is no leading '0' in the sub tables.

7.3.6 Progress Monitoring

The progress message during comparison is watched for error messages. Synchronization may fail if wrong tables are chosen or unwanted prefixing happened. The comparison may halt if more than half of the records didn't match. In case of errors check the data.

7.3.7 Intermediate Report Analysis

Once the comparison is over, an intermediate report is displayed. Check for the red ones. If more number of records rejected, enter Super Check stage and check the Anonymous Calls. Note the number of calls shown as Unknown. If more number of calls are left as unknown then check data. Enter command terminal and sort the unknown records date wise by [Suitable Query](#). Check whether more number of records are left unknown or rejected on a particular date.

7.3.8 Data Verification

The unknown calls must be verified manually in the verification stage. Fill up the form and press Execute. The records with Zero flag are shown. Press the BSNL button to check the record in the BSNL temporary sub table. Compare with the Private record and decide to accept or Reject the record and mark the radio button. When all the records are marked, press the Enter button at bottom of the page to write entries in the temporary table. Continue the process till you get the message verification completed. Then press Update to update the entries in the temporary table to Sub Table. In case of errors, press Discard to discard the temporary table.

7.3.9 Code Analysis

If the operator dealing is a Basic Service Operator, code analysis must be performed before report preparation. In the Code Analyzer window, fill up the form and enter the BSO Fixed levels. Give the code table name in **Database.Table** format. Execute code Analysis. In the command terminal pass [Suitable Query](#) to check the skipped calls. Analyze those calls and add missing entries if any in the code table.

7.3.10 Report Preparation

Enter Reporter window and fill the details. Execute reporter with Summary Report option. Check for warning message about verification. If warning, then go back to verification and complete it. Check the Summary Report. Execute the same with Detailed Report option. The Summary Report and Detailed Report may be sent to Accounts section for further calculation and settlement. The Reports can be saved in html format. The conqueror browser does not save the images when saving the html files like Internet Explorer and hence the BSNL logo should be saved separately by right clicking on the image in a folder named **images** with the html reports.

7.4 Work on server – Billing

The following is the procedure for Generating Bills for incoming calls of all types of POIs except MTNL.

7.4.1 Billing Table Creation

Create Billing Sub Tables for every Operator separately giving the Incoming Trunk Group Ids and period of Data. Check the tables for Data Integrity. If the Operator uses EWSD switch, the Called number may not have leading '0'. In such case prefix leading '0'. If the table is for VSNL check the checkbox, which enables the Home country calls inclusion.

7.4.2 Billing Code Analyser

Code Analyse the Billing Tables by giving correct POI types. Proper selection will save time. Check for any error message like **following code not available**. If valid codes are listed and not available in the code table, add those codes through '**add new code**' link in the tools. Do Code Analysis again.

7.4.3 Billing Reporter

Generate bill for each operator selecting appropriate POI type. Ensure that there is no error message like **Total count didn't match**. Attend to it. Recreate bill if necessary.

7.5 MTNL – Billing

The links related to MTNL are listed under Billing → MTNL. For both OG and IC traffic create sub tables, do code analysis and take report. The MTNL code tables are available in the code database in the name **mtnlby** and **mtnlnd**. Use correct MTNL code table for Mumbai and New Delhi. Select correct traffic type.

7.5 ILD OG – Billing

The links related to ILD OG are listed under Billing → ILD OG. Sub tables should be prepared for each ILD. The code table for ildog is separately available in the code table in the name **ildcodes**.

8 Problems Faced and Crossed

- OCB Records End time of a call
- EWSD Splits higher duration calls for every 1800 seconds by default
- Calling ID of OCB is restricted to 10 digits.
- There are records with calling_id as many zeros eg. 00000.
- Many single records are split into many in CMSP
- Many duplicate and triplicate records found in the PO CDR
- The called_id of PO is different from BSNL while other fields are same
- The duration recorded by PO is slightly higher.
- The format of PO data is dynamically changing
- The BSNL Clock and PO clock are not in phase.
- CDOT does not record date in every record and misinterpret the calls of cusp.
- ISD calls can not be identified using the CLI
- And many hidden problems

9 Support and contact information

To Get the Software:

<p>Sri. R. GANESAN General Manager, NMS Southern Telecom Region Chennai Mobile: 9444056878</p>

For Technical Support:

<p>Sri. T.UMAPATHI ANAND Junior Telecom Officer New Tech Installation Coimbatore. Mobile: 9443350005</p>	<p>Sri. ARUL SELVANAR Junior Telecom Officer (DTAX) DTAX Maintenance Karaikudi Phone: 04565-220000</p>
---	---

<p>Sri. R.KASTHURI Sub Divisional Engineer (IUC) Network Management System Southern Telecom Region Chennai. Mobile: 9444120999</p>	<p>Sri. S.BALASUBRAMANIAM Sub Divisional Engineer (Tech) Network Management System Southern Telecom Region Chennai. Mobile: 04420019038</p>
---	--

Appendix A

A. IUCIT Standard Table Structures

BSNL Master Table (Non EWSD)

Field Name	Field Type
date	Date
time	Time
calling_id	Varchar(20)
called_id	Varchar(20)
duration	Int(5)
units	Int(5)
otgp_id	Varchar(10)
itgp_id	Varchar(10)

BSNL Master Table (EWSD)

Field Name	Field Type
date	Date
time	Time
calling_id	Varchar(20)
called_id	Varchar(20)
duration	Int(5)
units	Int(5)
otgp_id	Varchar(10)
itgp_id	Varchar(10)
conn_id	Bigint(12)

Private Master Table

Field Name	Field Type
date	Date
time	Time
calling_id	Varchar(20)
called_id	Varchar(20)
duration	Int(5)

Code Table

Field Name	Field Type
code*	Varchar(6)
slab	Tinyint(1)
ppslab	Tinyint(1)
cat	Varchar(4)
ccat	Varchar(4)
len	Tinyint(1)
I2	Char(3)
I3	Char(3)

* → Primary Key

subcode

Field Name	Field Type
code*	Char(3)
cat	Char(3)
len	Tinyint(1)

* → Primary Key

ildcodes

Field Name	Field Type
code*	Char(8)
country name	Char(30)
cat0	Char(3)
cat1	Char(3)
cat2	Char(3)
cat3	Char(3)

* → Primary Key

mtnlby

Field Name	Field Type
code*	Char(6)
cat	Char(3)
len	Tinyint(1)

* → Primary Key

mtnlnd

Field Name	Field Type
code*	char(6)
cat	char(3)
len	tinyint(1)

* → Primary Key

B. Values of cat and slab fields in the Code Table**The values of 'slab'**

Value	Meaning
1	0 – 50 Km
2	51 – 200 Km
3	201 – 500 Km
4	Above 500 Km

The Values of 'cat' Phase I

Value	Meaning
IC	Intra Circle
OC	Other Circle
BF	BSNL Fixed
BM	BSNL Mobile

The values of 'cat' Phase II

Value	Meaning
IC	Intra Circle
OC	Other Circle
LO	Local (same SDCA)
BF	BSNL Fixed
BM	BSNL Mobile
PF	Private Fixed
PM	Private Mobile
PC	Private Cell
BC	BSNL Cell
BE	BSNL Emergency
OF	Other Fixed

C. IUCIT Flags (Reconciliation)

Sl.No	Flag	Meaning
1	0	Not Verified
2	1	Matched
3	A	Accepted Call
4	C	No CLI
5	D	Duplicated Call
6	H	Higher Duration
7	L	Lower Duration
8	I	ISD Call
9	P	PBX Call
10	R	Redirected Call
11	S	Split Call
12	X	Rejected Call
13	Z	Zero Duration

Appendix B

Important MySQL Commands (assuming the table name as 'airtel')

```
mysql> select date,count(*) from airtel group by date;

mysql> select date,count(*) from airtel where flag ='0' group by date ;

mysql> select left(time,2) as hor, count(*) from airtel where date = '2004-05-02'
group by hor;

mysql> update airtel set flag='A' where flag='0' and calling_id like '4222%';

mysql> select flag,count(*) from airtel group by flag;

mysql> select sum(duration) as duration, sum(duration)/60 as MOU from airtel ;

mysql> update airtel set called_id = concat('452',called_id);

mysql> update airtel set calling_id = trim(leading '0' from calling_id);

mysql> delete from airtel where duration < 2;

mysql> insert into airtel select * from at_anex;

mysql> create table mytable(dt date, tm time, sales int(5), name varchar(20));

mysql> select count(*) from airtel where slab is NULL and calling_id="";

mysql> update airtel set flag='A' where flag!='D' and flag!='X';

mysql> truncate table airtel;

mysql> alter table airtel add id varchar(15);

mysql> alter table iuccode add primary key(code);

mysql> describe table iuccode;

mysql> show keys from iuccode;

mysql> select * from airtel where calling_id like '4__2%';

mysql> select left(calling_id,3) as code, count(*) from airtel group by code;
```

for More command refer MySQL Document.