



Tally.ERP9 – Fusion TallyJDBC/TallyAPI /JDBC - Java  
[com.cevious.tally.jdbc](http://com.cevious.tally.jdbc)

The Tally logo is the word "Tally" written in a bold, black, cursive script font. The letters are thick and have a slight shadow or outline effect, giving it a three-dimensional appearance.

**POWER OF SIMPLICITY**

## Contents

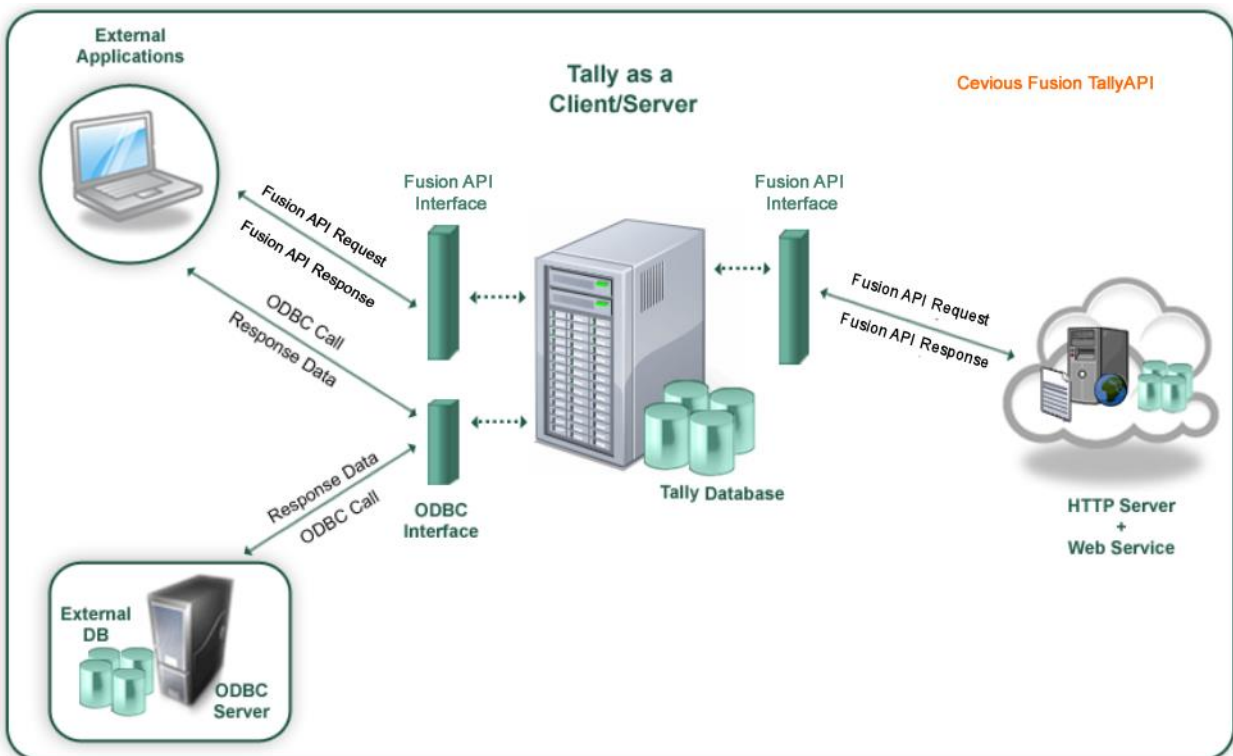
1	Introduction .....	2
2	Need and Benefits of Integration.....	2
3	Tally Interfaces – For Integration.....	3
3.1	Tally ODBC Interface (Read Only) .....	3
3.2	Tally API/JDBC Interface (Read and Write) .....	3
4	Integration Using Tally API Interface .....	4
4.1	Tally.ERP 9 as a Server – Using External application as Front End.....	4
4.2	TallyAPI /JDBC Configuration using Java.....	4
4.2.1	TallyAPI /JDBC Required jar .....	4
4.2.2	TallyAPI /JDBC ITallyData Class Structure .....	5
4.2.3	TallyAPI /JDBC TTransaction Class Diagram .....	5
4.2.4	TallyAPI /JDBC Application setup & Initialization.....	5
4.2.5	Case Study I - Create Ledger from Java.....	6
4.2.6	Case Study II - Create Ledger with auto fill AccountGroup by ODBC.....	8
4.2.7	Case Study III – Voucher Creation using TallyAPI/JDBC .....	10

# 1 Introduction

Large and medium sized businesses use disparate applications to run their business and one of the major areas that need to converge amongst these applications is the Accounting, Financial and Inventory information. Tally being the default accounting, Inventory and Statutory Compliance software used by enterprises in these segments. Therefore the need arises to discuss on the Integration Capabilities of Tally. Integration Solutions are designed to ensure that the existing investments in Software (ERP, Legacy and other Enterprise systems) remain intact by seamlessly integrating information with new systems, technologies and custom applications within the enterprise, as well as with companies with whom the business deals with.

# 2 Need and Benefits of Integration

To meet the challenges of the new business environment, information systems need to communicate with each other as seamlessly as possible, provide right-time visibility of transactions across the entire enterprise and be flexible enough to accommodate the changing structure of the business. When more and more information needs to be shared across traditional business boundaries, the way you integrate your systems and processes is rapidly becoming one of the most important priorities in business today. The following figure gives a complete perspective on the overall Integration Capabilities of Tally.ERP 9



### 3 Tally Interfaces – For Integration

Tally communicates with the external world mainly using two Interfaces.

- Tally ODBC Interface (Read Only)
- Tally API/ JDBC Interface (Read and Write)

#### 3.1 Tally ODBC Interface (Read Only)

ODBC (Open Database Connectivity) makes it possible to access data from any application, regardless of which Database Management System (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver between an application and the DBMS. The purpose of this layer is to translate the application's database queries into commands that the DBMS can understand. For this to function, both the application and the DBMS must be ODBC compliant i.e., the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.

Tally provides the ODBC Interface which makes it possible for applications to talk to Tally Database. By using this interface, external applications will be able to retrieve data from Tally. Tally acts as a Server delivering Data to external applications Using the ODBC Interface, Tally.ERP 9 can make ODBC calls to an External Database and retrieve data from them. In such a case Tally acts as a Client to pull Data from disparate Data Sources. This data can be consumed in Tally as per requirement

#### 3.2 Tally API/JDBC Interface (Read and Write)

API (Application Program Interface) is the standard for information exchange with external systems. Tally.ERP 9 supports standardized message formats for Read/Write. Tally.ERP 9 can communicate with any environment capable of Writing and reading with TallyAPI /JDBC.

Tally can act as an DB Server capable of receiving an API Request and responding with an API Response. The entire Tally Data can be made available to the requesting application. It is also possible for the application to store data into Tally Database.

Using the same interface, Tally has the capability to interact with a Web Service delivering Data over HTTP. In this scenario, Tally behaves as a client retrieving and storing data into an external database. The Web Service capable of handling Tally Read/Write serves as a layer between Tally and External Database.

## 4 Integration Using Tally API Interface

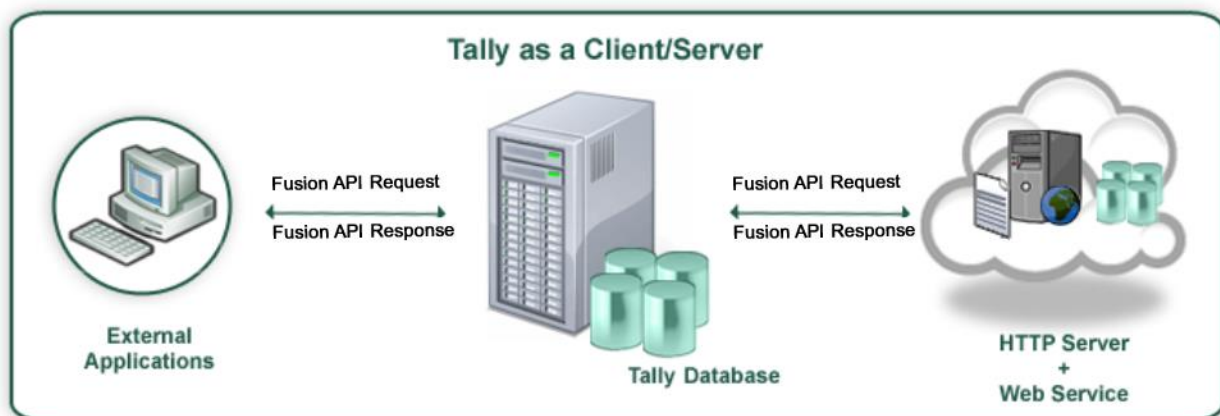
On the completion of this chapter you will be able to

Understand the functionality of Tally as a Server using External applications

Introduction Tally.ERP 9 has supported integration with web scripting languages such as ASP/Perl/PHP and other languages like VB or any environment capable of supporting API and HTTP. Integration with these products is possible as API import and export capability is built into TallyAPI /JDBC functionality

In fact, Tally.ERP 9 delivers most of the functionalities of Web Services provided by Micro-soft's.NET framework. All Tally.ERP 9 data is accessible to any number of potentially disparate systems through the use of Internet standards such as API and HTTP.

In other words, Tally.ERP 9 can communicate with any environment capable of sending and receiving API over HTTP. This chapter explains how Tally.ERP 9 will act as a server/client while it is connecting to external applications. The following figure shows the API OOPS Messaging Format through external application, acting Tally.ERP 9 as a Server/Client.



### 4.1 Tally.ERP 9 as a Server – Using External application as Front End

Data can be accessed from Tally.ERP 9 once the connection is established between Tally.ERP 9 and other external applications. Here we are using Data can be posted from Java to Tally.ERP 9 through API Interface. The existing Tally.ERP 9 data can be altered and deleted from Java. Let us discuss some scenarios for using external application as front end.

### 4.2 TallyAPI /JDBC Configuration & Sample using Java

TallyJDBC Sample Configure using NetBeans IDE 8.0.1

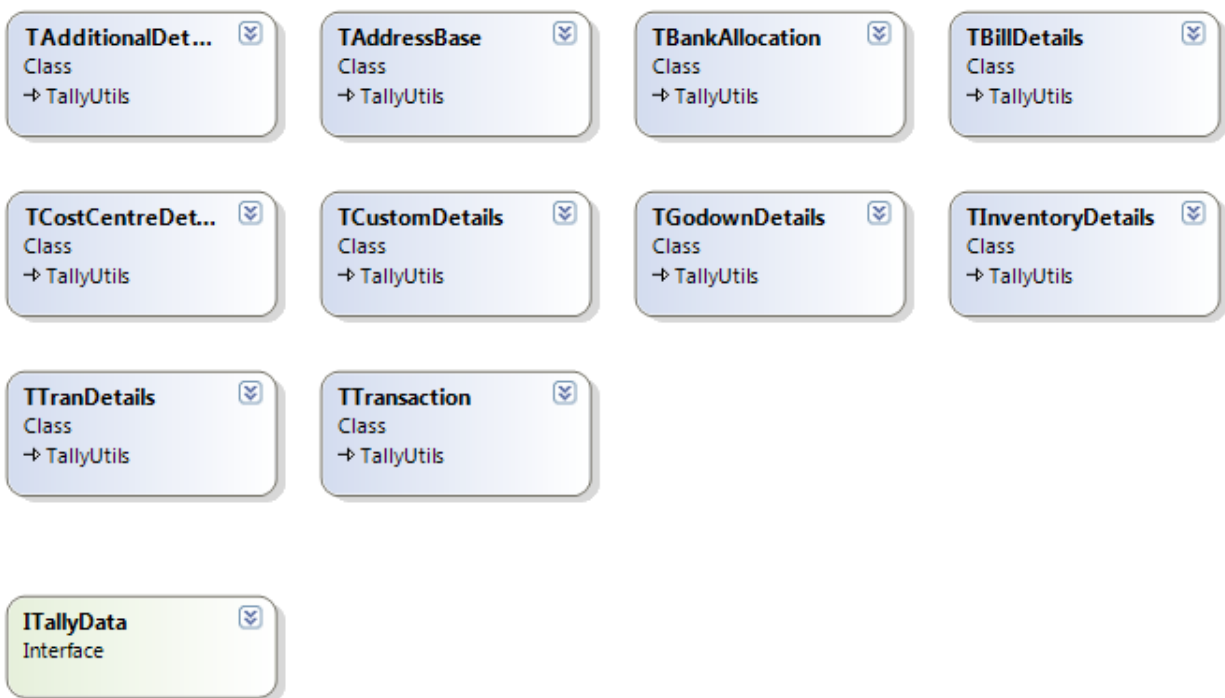
#### 4.2.1 TallyAPI /JDBC Required jar

- com.cevious.tally.jdbc.jar

## 4.2.2 TallyAPI /JDBC ITallyData Class Structure

InterFace	Base Class	Drived Class	Description
ITallyData			Tally Data Interface
	TallyUtils		Tally Data BaseClass
		Tledger	Ledger master
		TAccountGroup	Account Group
		TCostCategory	Cost Category
		Tproduct	Product
		TProductGroup	Product Group
		TProductCategory	Product Category
		Tgodown	Godown
		Tunit	Unit
		Ttransaction	Tally Transaction

## 4.2.3 TallyAPI /JDBC TTransaction Class Diagram



## 4.2.4 TallyAPI /JDBC Application setup & Initialization

```

/// <summary>
    /// The main entry point for the application.
    /// </summary>
public class TallyJDBCTest {

    public static void main(String[] args) {

        TallyApplication tallyApp=new TallyApplication();
    }
}
    
```

```
//Default Conection is localhost:9000

tallyApp.setConnection(new TallyConnection("localhost:9000"));

tallyApp.setCompany("Fusion");

TallyController tallyController=new TallyController(tallyApp);

//TallyException

try{

    if(tallyController.IsCompanyOpen())

    {

        //Transaction & Master wrapper code

        //tallyController.Post(CreateJournalVoucher(VoucherType.Journal));

        //System.out.println("Voucher Post");

    }

}

}catch(TallyException te){

    System.out.println( te.getMessage());

}

}catch(Exception ex){

    System.out.println( ex.getMessage());

}

}
```

#### 4.2.5 Case Study I - Create Ledger from Java

A Company Demo Software needs to design an interface for entering their Ledger master with parent (account group). At the end of Master Entry, the same needs to be posted to Tally.ERP 9.

The following console code has been designed for Ledger Master Entry

Tally API/LDBC OOPS fragment get generated and the same is being posted to Tally.ERP 9 running in a defined port.

#### Java Interface Code for Master Creation

```
/// <summary>
    /// Integration Sample for Ledger master Creation Tally By manual entering
    ///Ledgername, Group

import com.cevious.tally.jdbc.*;
```

```
import com.cevious.tally.jdbc.datatable.*;
import java.util.*;
import java.util.UUID;
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Fusion
 */
public class TallyJDBCMaster {
    public static void main(String[] args) {
        TallyApplication tallyApp=new TallyApplication();
        //Default Connection is localhost:9000
        tallyApp.setConnection(new TallyConnection("localhost:9000"));
        tallyApp.setCompany("Fusion");
        TallyController tallyController=new TallyController(tallyApp);
        //TallyResponseException,NullReferenceException,TallyException
        try{
            if(tallyController.IsCompanyOpen())
            {
                TLedger tLedger=CreateLedger("InderJeet Test Ledger","Sundry
Debtors",true);
                tallyController.Post(tLedger);
                tLedger=CreateLedger("InderJeet JDBC Ledger","Sundry Debtors2",true);
                tallyController.Post(tLedger);
                System.out.println("Ledger Post");
            }

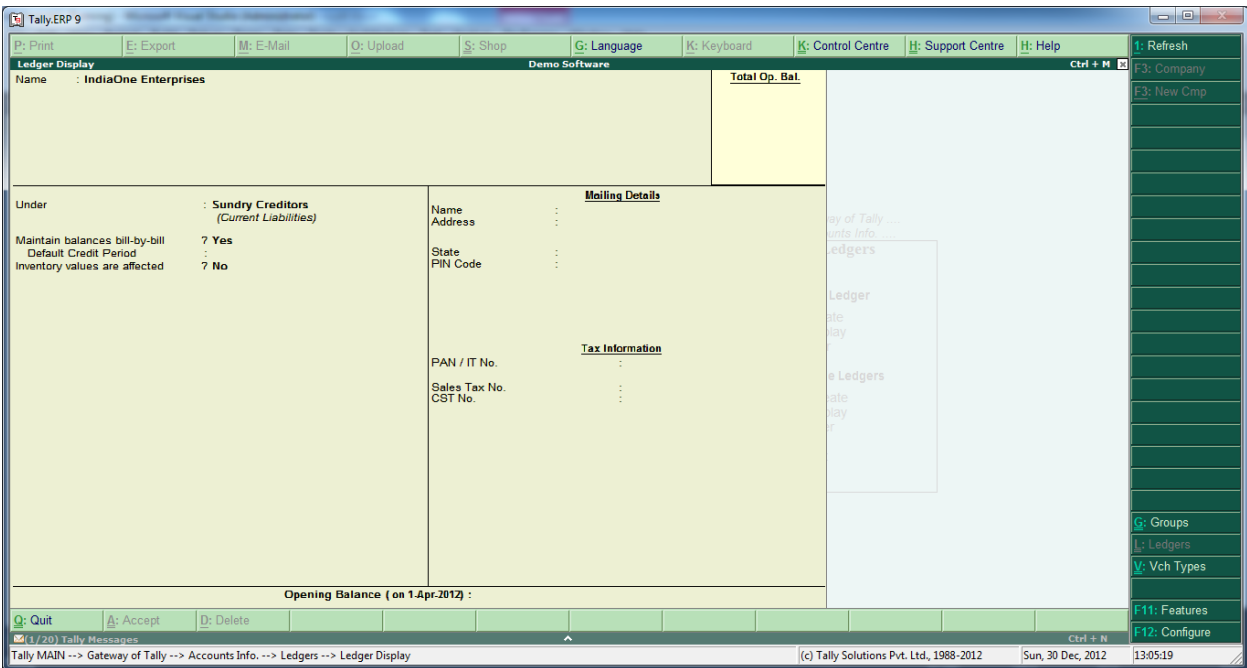
            }catch(TallyException te){
                System.out.println( te.getMessage());
            }catch(Exception ex){
                System.out.println( ex.getMessage());
            }
        }

        public static TLedger CreateLedger(String name, String group, Boolean isBill){
            TLedger tLedger=new TLedger(name,new TAccountGroup(group));
            tLedger.IsBillByBill=isBill;

            return tLedger;
        }
    }
}
```



On Creating Ledger, Show below:



The above Tally.ERP9 screen displays the Ledger master created from the external Interface.

#### 4.2.6 Case Study II - Create Ledger with auto fill AccountGroup by ODBC

A Company Demo Software needs to design an interface for entering their Ledger master with parent (account group). At the end of Master Entry, the same needs to be posted to Tally.ERP 9.

Here All AccountGroup display by Tally JDBC using API SQL Request

The following interface has been designed for Ledger Master Entry

The above Interface is designed for fetch all Accountgroup on form load from Tally JDBC use SQL API fragment and get response. On selection of a account group

#### Java Code for SQL API Request

This code is already define in above TallyJDBC common library section.

```

/// <summary>
/// Ensure TallyAPI /JDBC Establish in Tally & Company should Open
    
```

```
    /// On Form Loading Establishing Connection with tally & Fetch Account Group to
console
    import com.cevious.tally.jdbc.*;
import com.cevious.tally.jdbc.datatable.*;
import java.util.*;
import java.util.UUID;
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Fusion
 */
public class TallyJDBCDataFetch {
    public static void main(String[] args) {
        TallyApplication tallyApp=new TallyApplication();
        //Default Connection is localhost:9000
        tallyApp.setConnection(new TallyConnection("localhost:9000"));
        tallyApp.setCompany("Fusion");
        TallyController tallyController=new TallyController(tallyApp);
        //TallyResponseException,NullReferenceException,TallyException
        try{
            if(tallyController.IsCompanyOpen())
            {
                //TLedger tLedger=CreateLedger("JDBC Test Ledger","Sundry Debtors2");
                //tallyController.Post(tLedger);
                FetchData(tallyController,TallyTable.Ledger);
            }

        }catch(TallyException te){
            System.out.println( te.getMessage());
        }catch(Exception ex){
            System.out.println( ex.getMessage());
        }
    }

    public static TLedger CreateLedger(String name, String group, Boolean isBill){
        TLedger tLedger=new TLedger(name,new TAccountGroup(group));
        tLedger.IsBillByBill=isBill;
        return tLedger;
    }

    public static void FetchData(TallyController controller, TallyTable table){
        String sql="Select $Name,$Parent from "+table;
        DataTable dataTable=controller.ExecutesQL(sql);
        for(DataColumn dt:dataTable.Columns()){
            System.out.println("Columns : " +dt.getColumnName());
        }

        for(DataRow dt:dataTable.Rows()){
            System.out.println("Row : "
+dt.getValue("$NAME")+"=>" +dt.getValue("$Parent"));
        }
    }
}
```

```
}
```

The above Tally.ERP9 screen displays the Ledger master created from the external Interface and using Tally AccountGroup.

#### 4.2.7 Case Study III – Voucher Creation using TallyAPI/JDBC

A Company Demo Software needs to design an interface for entering their receipt vouchers and altering the same, if required. At the end of Voucher Entry, the same needs to be posted to Tally.ERP 9.

Data being posted to Tally.ERP 9 running in a defined address:port

#### Java Code for Voucher Creation;

```
    /// <summary>
    /// Create Button to Post API request to tally and Create request voucher
    /// </summary>
import com.cevious.tally.jdbc.*;
import com.cevious.tally.jdbc.datatable.*;
import java.util.*;
import java.util.UUID;
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Fusion
 */
public class TallyJDBCTest {
    public static void main(String[] args) {
        TallyApplication tallyApp=new TallyApplication();
        //Default Connection is localhost:9000
        tallyApp.setConnection(new TallyConnection("localhost:9000"));
        tallyApp.setCompany("Fusion");
        TallyController tallyController=new TallyController(tallyApp);
        //TallyException
        try{
            if(tallyController.IsCompanyOpen())
            {
                tallyController.Post(CreatePaymentVoucher(VoucherType.Payment));
                tallyController.Post(CreateReceiptVoucher(VoucherType.Receipt));
                tallyController.Post(CreateJournalVoucher(VoucherType.Journal));
                System.out.println("Voucher Post");
            }

        }catch(TallyException te){
            System.out.println( te.getMessage());
        }catch(Exception ex){
            System.out.println( ex.getMessage());
        }
    }
}
```

```
public static TLedger CreateLedger(String name, String group, Boolean isBill){
    TLedger tLedger=new TLedger(name,new TAccountGroup(group));
    tLedger.IsBillByBill=isBill;
    return tLedger;
}
```

## Java Code for Payment Voucher;

```
public static TTransaction CreatePaymentVoucher(VoucherType nature){
    TTransaction voucher=new TTransaction(nature);
    // set VoucherType other than original type
    voucher.setVoucherType("Bank Payment");
    // Tally transaction ID should be unique every time.
    // Your ID will help you update Tally data every time.
    // New ID to Create Tally transaction Only One Time not for update.
    voucher.setTOid(UUID.randomUUID().toString());
    Date date=IAFModule.parseDate("01-04-2014", "dd-MM-yyyy");
    voucher.setDate(date);
    voucher.setNarration("Java JDBC Test");
    voucher.setVoucherNumber("1");
    voucher.setReferences("1");
    //Transaction Details ledger transaction entry
    String party="Inderjeet Delhi Party";
    TTranDetails tranDetail=new TTranDetails(voucher,
        CreateLedger(party,"Sundry Debtors", true),
        71000,"Dr");
    //// Auto Create Ledger in Tally if IsUpdateMaster is True
    tranDetail.setIsUpdateMaster(true);
    if(tranDetail.getLedgers().IsBillByBill)
    {
        TBillDetails tBill=new TBillDetails(
            tranDetail,
            voucher.getVoucherNumber(),
            tranDetail.getAmount());
        tranDetail.getBillDetails().add(tBill);
    }
    voucher.getTranDetails().add(tranDetail);
    tranDetail=new TTranDetails(voucher,
        CreateLedger("HDFC Bank","Bank Accounts", false),
        71000,"Cr");
    TBankAllocation tBank= new TBankAllocation(tranDetail);
    tBank.setBankName("HDFC Bank");
    tBank.setParty(party);
    tBank.setInstrumentNumber("327839");
    tBank.setFavouringName("Sample Party");
    tranDetail.getBankAllocations().add(tBank);
    voucher.getTranDetails().add(tranDetail);

    //for(TTranDetails ttd:voucher.getTranDetails())
    //    System.out.println(voucher.getDefaultDRCR() + ">" +
ttd.getLedgers().getName());
    return voucher;
}
```

## Java Code for Receipt Voucher

```
public static TTransaction CreateReceiptVoucher(VoucherType nature){
    TTransaction voucher=new TTransaction(nature);
    // set VoucherType other than original type
    //voucher.setVoucherType("Bank Payment");
    // Tally transaction ID should be unique every time.
    // Your ID will help you update Tally data every time.
    // New ID to Create Tally transaction Only One Time not for update.
    voucher.setTOid(UUID.randomUUID().toString());
    Date date=IAFModule.parseDate("01-04-2014", "dd-MM-yyyy");
    voucher.setDate(date);
    voucher.setNarration("Java JDBC Test");
    voucher.setVoucherNumber("1");
    voucher.setReferences("1");
    //Transaction Details ledger transaction entry
    String party="Inderjeet Delhi Party";
    TTranDetails tranDetail=new TTranDetails(voucher,
        CreateLedger(party,"Sundry Debtors", true),
        71000,"Cr");
    //// Auto Create Ledger in Tally if IsUpdateMaster is True
    tranDetail.setIsUpdateMaster(true);
    if(tranDetail.getLedgers().IsBillByBill)
    {
        TBillDetails tBill=new TBillDetails(
            tranDetail,
            voucher.getVoucherNumber(),
            tranDetail.getAmount());
        tranDetail.getBillDetails().add(tBill);
    }
    voucher.getTranDetails().add(tranDetail);
    tranDetail=new TTranDetails(voucher,
        CreateLedger("HDFC Bank","Bank Accounts", false),
        71000,"Dr");
    TBankAllocation tBank= new TBankAllocation(tranDetail);
    tBank.setBankName("HDFC Bank");
    tBank.setParty(party);
    tBank.setInstrumentNumber("327839");
    tBank.setFavouringName("Sample Party");
    tranDetail.getBankAllocations().add(tBank);
    voucher.getTranDetails().add(tranDetail);

    //for(TTranDetails ttd:voucher.getTranDetails())
    //    System.out.println(voucher.getDefaultDRCR() + ">"+
    ttd.getLedgers().getName());
    return voucher;
}
```

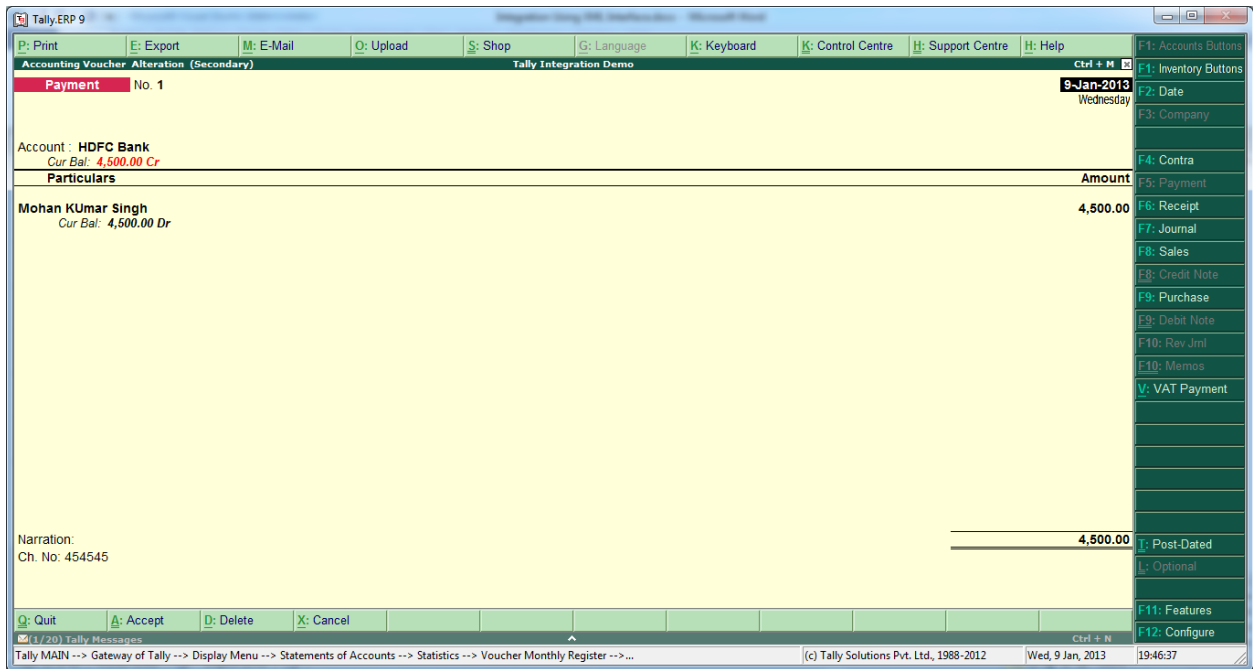
## Java Code for Journal Voucher

```
public static TTransaction CreateJournalVoucher(VoucherType nature){
    TTransaction voucher=new TTransaction(nature);
    // set VoucherType other than original type
    //voucher.setVoucherType("Payment");
    // Tally transaction ID should be unique every time.
    // Your ID will help you update Tally data every time.
    // New ID to Create Tally transaction Only One Time not for update.
```

```
voucher.setT0id(UUID.randomUUID().toString());
Date date=IAFModule.parseDate("01-04-2014", "dd-MM-yyyy");
voucher.setDate(date);
voucher.setNarration("Java JDBC Test");
voucher.setVoucherNumber("1");
voucher.setReferences("1");
//Transaction Details ledger transaction entry
String party="JDBC Party";
TTranDetails tranDetail=new TTranDetails(voucher,
    CreateLedger(party,"Sundry Debtors", true),
    81000,"Cr");
//// Auto Create Ledger in Tally if IsUpdateMaster is True
tranDetail.setIsUpdateMaster(true);
if(tranDetail.getLedgers().IsBillByBill)
{
    TBillDetails tBill=new TBillDetails(
        tranDetail,
        voucher.getVoucherNumber(),
        tranDetail.getAmount());
    tranDetail.getBillDetails().add(tBill);
}
voucher.getTranDetails().add(tranDetail);
tranDetail=new TTranDetails(voucher,
    CreateLedger("Office Expenses","Indirect Expenses", false),
    81000,"Dr");
TCostCentreDetails tcc= new TCostCentreDetails(tranDetail);
tcc.setCostCentre(new TCostCentre("Office Travel",new TCostCategory("Primary Cost
Category"))));
tcc.setAmount(81000);
tranDetail.getCostCentreDetails().add(tcc);
voucher.getTranDetails().add(tranDetail);

//for(TTranDetails ttd:voucher.getTranDetails())
//    System.out.println(voucher.getDefaultDRCR() + ">" +
ttd.getLedgers().getName());
return voucher;
}
}
```

The above Tally.ERP 9 Screen displays the Voucher which has been altered from an external interface application based on VoucherType, VoucherNumber And Date. Narration show “Voucher Modified”



The above Tally.ERP 9 Screen displays the Voucher which has been created from an external interface