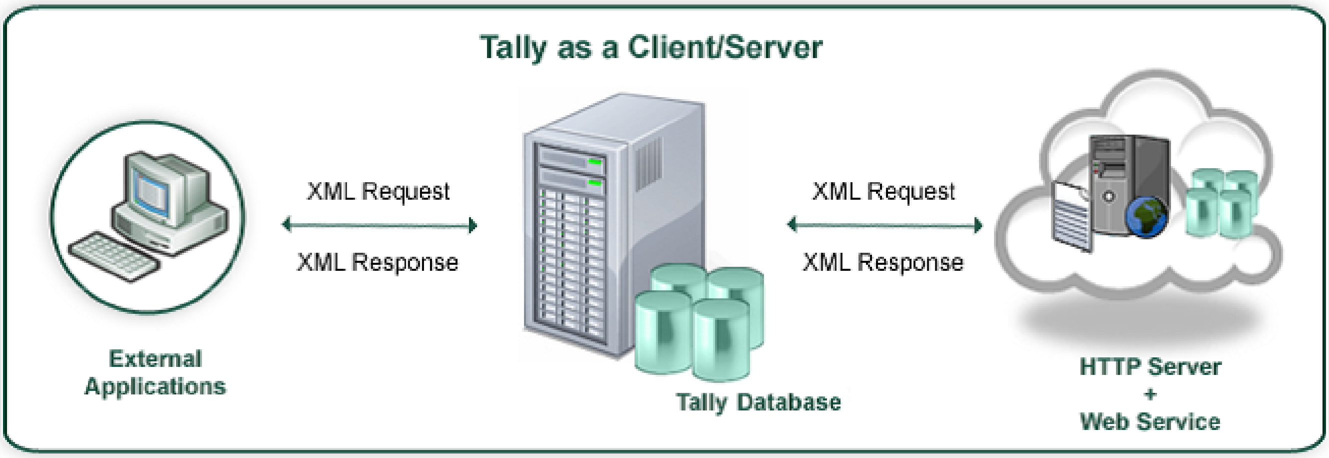
Integration With TallyPrime

Introduction

Tally can communicate with any environment capable of sending and receiving XML over HTTP. TallyPrime supports integration with Third Party Applications (like web scripting languages such as ASP/Java/PHP and applications built, which supports XML and HTTP).

Tally can act as an HTTP Server capable of receiving an XML Request and responding with an XML 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 directly without using Tally User Interface.

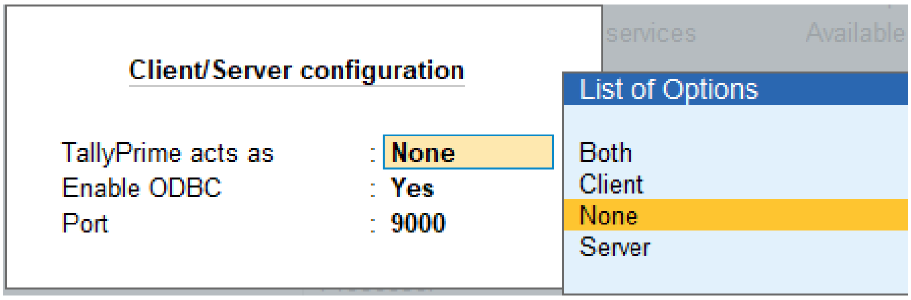


Pre-requisites for Integration

1. TallyPrime must be running on one of the port (For example: 9000)

-
    - Open TallyPrime
    - Go to **Exchange** > **Data Synchronization**

The Data Synchronization Configuration screen appears.



Client/Server configuration

TallyPrime acts as : None
Enable ODBC : Yes
Port : 9000

Configure the parameters as given below:

Configuration	Value
TallyPrime acting as	Both / Server / Client / None

Enable ODBC	Yes/No
Port	9000 (desired port number)

2. Company must be loaded in TallyPrime

To select a company Go to **Company > Select**, it lists the companies created, select the company and load it.

HTTP Request Format

Every request send to Tally from a third party must contain the information shared below. Tally processes the HTTP request and sends response in XML format itself.

HTTP Request Headers	Information
HTTP method	POST HTTP/1.1
Host	Localhost:9000
Content-Type	text/xml,UTF-8,UTF-16,ASCII

HTTP Body	Information
Content-Type	text/xml, UTF-8, UTF-16, ASCII
Data	XML request content

XML Request and Response

A basic XML structure will have an Envelope enclosed tags with Header and Body tags. This basic XML structure will be present in both XML request and response.

```
<ENVELOPE>
    <HEADER> . . . </HEADER>
```

```
        <BODY> . . . </BODY>

    </ENVELOPE>
```

Requests

XML request structure depends on the type of Tally request actions export, import, and execute. A basic XML request will be as follows:

```
    <ENVELOPE>

        <HEADER>

            <VERSION>Version Number</VERSION>

            <TALLYREQUEST>Request Type</TALLYREQUEST>

            <TYPE>Information Type</TYPE>

            <SUBTYPE>Sub Type</SUBTYPE>

            <ID >Identifier</ID>

        </HEADER>

        <BODY>

        <DESC>

          <STATICVARIABLES>Static Variables Specification</STATICVARIABLES>

          <REPEATVARIABLES>Repeat Variables Specification</REPEATVARIABLES>

          <FETCHLIST> Fetch Specification</FETCHLIST>

          <FUNCPARAMLIST> Parameter specification in case of function type

          </FUNCPARAMLIST>

            <TDL>   TDL Information     </TDL>

        </DESC>

            <DATA>  Data (if applicable) </DATA>

        </BODY>

    </ENVELOPE>
```

Response

TallyPrime processes requests and send response accordingly. A basic XML request will be as follows:

```
    <ENVELOPE>

    <HEADER>

     <VERSION>Version Number</VERSION>

     <STATUS>-1/0/1</STATUS>
```

```
</HEADER>

<BODY>

 <DESC>

    <STATICVARIABLES> Static Variables Specification </STATICVARIABLES>

 </DESC>

 <DATA> Data retrieved (if available)</DATA>

</BODY>

</ENVELOPE>
```

For detailed information about the tags used in request and response, refer here.

How to send a request from Third Party Applications to TallyPrime

To send a HTTP request to Tally application following details are required:

- **URL** – As a prerequisite, TallyPrime must be running on one of the port. For example, if the XML request is sent to TallyPrime on the port 9000 on the local system, the URL "http://localhost:9000" is specified.
- **Headers** – Use the standard HTTP headers

  HTTP method: POST HTTP/1.1

  Host: Localhost: 9000

- **Content-Type**: Tally supports the following content types of XML SOAP requests
  - UTF-8

    Unicode Text Format – 8 (UTF–8) is the default response content-type of Tally. Request made with content type "text/xml" or "UTF–8" will get the Tally response in UTF–8 format.

    For example, Content-type:"UTF-8" or Content-type: "text/XML"

- - UTF-16

    Unicode Text Format – 16 (UTF-16) is an extended text format of UTF-8. UTF-16 supports special characters, such as currency symbols Rupee, Euro etc.

    The request content need to be encoded as Unicode encoding to post to Tally, so that Tally will respond in Unicode UTF-16 format.

    For example, Content-type:"UTF-16"

- - ASCII

    ASCII (American Standard Code for Information Interchange). Specify the content type as ASCII and encode the request content as ASCII encoding to post to Tally, so that Tally respond in ASCII format.

For example, Content-type:"ASCII"

POST/Body text – Post/body text will be Tally XML request format. Following is a XML request format to extract the list of ledgers in Tally.

```
<ENVELOPE>

    <HEADER>

        <VERSION>1</VERSION>

        <TALLYREQUEST>EXPORT</TALLYREQUEST>

        <TYPE>COLLECTION</TYPE>

        <ID>List of Ledgers</ID>

    </HEADER>

    <BODY>

    <DESC>

        <STATICVARIABLES>

            <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>

        </STATICVARIABLES>

    </DESC>

    </BODY>

</ENVELOPE>
```

Set the above parameters and post a request to TallyPrime, it will respond with list of ledgers present in that company.

Here are some of the sample HTTP Post request codes for reference, Request XML need to be replaced with the Tally XML request content.

Samples

Sample PHP Request

You can include the following code in PHP script to initiate a request

```
   <?php

   $request = new HttpRequest();

   $request->setUrl('http://localhost:9000/');

   $request->setMethod(HTTP_METH_POST);

   $request->setHeaders(array(

     'cache-control' => 'no-cache',

     'content-type' => 'text/xml'));

   $request->setBody(' Request XML ');
```

```
try {   $response = $request->send();
  echo $response->getBody();
} catch (HttpException $Msg_ex) { echo $Msg_ex;   }    ?>
```

Sample Java Request

You can include the following code in a Java class to initiate a request

```
OkHttpClient client = new OkHttpClient();
MediaType mediaType = MediaType.parse(“text/xml “);
RequestBody body = RequestBody.create(mediaType, “Request XML”);
Request request = new Request.Builder()
  .url(“http://localhost:9000/”)
  .post(body)
  .addHeader(“content-type”, “text/xml”)
  .addHeader(“cache-control”, “no-cache”)
  .build();
Response response = client.newCall(request).execute();
```

Sample C# Request

Include the following code in a C# class to initiate a request

```
var client = new RestClient(“http://localhost:9000/”);
var request = new RestRequest(Method.POST);
request.AddHeader(“cache-control”, “no-cache”);
request.AddHeader(“content-type”, “text/xml”);
request.AddParameter(“text/xml”,” Request XML “, ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Sample VB.Net Request

Include the following code in a VB class to initiate a request

```
Dim data As String = RequestXML
Dim url As String = “http://localhost:9000“
Dim myReq As HttpWebRequest = WebRequest.Create(url)
Dim encoding As New UnicodeEncoding
Dim buffer() As Byte = encoding.GetBytes(data)
myReq.AllowWriteStreamBuffering = False
myReq.Method = “POST”

myReq.ContentType = “UTF-16”

myReq.ContentLength = buffer.Length

Dim post As Stream = myReq.GetRequestStream

post.Write(buffer, 0, buffer.Length)

post.Close()

Dim myResponse As HttpWebResponse = myReq.GetResponse

   If myResponse.StatusCode = HttpStatusCode.OK Then

     Dim responsedata As Stream = myResponse.GetResponseStream

     Dim responsereader As StreamReader = New StreamReader

      (responsedata, System.Text.Encoding.Unicode, True)

      response = responsereader.ReadToEnd

      Resp_find.Text = response

  End If
```

Accounting Masters

Masters are the most important entity to record transactions in Tally. The accounting masters primarily consists of Group, Ledger and Voucher Types. The ledger is the actual accounting head to identify your transactions and generate all accounting reports.

The following table helps to understand the accounting masters and its purpose:

Master	Information
Group	Group is a collection of ledgers
Ledger	Ledger is the actual accounting master used for recording transactions and generating account reports
Voucher Type	Voucher Type used to categorize the different nature of transactions, provide specific voucher number and configure its behaviour while recording the transactions

Group

Group is a collection of ledgers of the same nature. Account groups are maintained to determine the hierarchy of Ledger Accounts, which is helpful in determining and presenting meaningful and compliant reports. Click here for the steps to create a group manually in Tally.

Tally provides the flexibility to the user for organizing the chart of accounts. The following XML request can be send to Tally to create a group "North Zone Debtors" under predefined group Sundry Debtors:

```
<ENVELOPE>

<HEADER>

    <TALLYREQUEST>Import Data</TALLYREQUEST>

</HEADER>

<BODY>

<IMPORTDATA>

      <REQUESTDESC>

        <REPORTNAME>All Masters</REPORTNAME>

      </REQUESTDESC>

      <REQUESTDATA>

        <TALLYMESSAGE xmlns:UDF="TallyUDF">

            <GROUP Action="Create">

              <NAME>North Zone Debtors</NAME>

              <PARENT>Sundry Debtors</PARENT>

            </GROUP>

        </TALLYMESSAGE>

      </REQUESTDATA>

</IMPORTDATA>

</BODY>

</ENVELOPE>
```

Ledger

The **Ledgers** are commonly known as general ledger in accounting standards. Ledgers provides the summary of recorded transactions and helps to generate accounting reports. A ledger in Tally must be created under a group. Click here for the steps to create a ledger manually in Tally.

The following table helps to understand the XML schema with the minimal tags required for creating a ledger.

Tag	Data type	Is mandatory	Description

LEDGER		Yes	Opening tag
NAME	String	Yes	To provide the name of the Ledger
PARENT	String	Yes	This should contain a group name from list available in Tally. We can also create our own groups and classify ledger under that group subject to the group is created prior to ledger creation.

The below given XML request creates a ledger in Tally

```
<ENVELOPE>
<HEADER>
    <TALLYREQUEST>Import Data</TALLYREQUEST>
</HEADER>
<BODY>
<IMPORTDATA>
      <REQUESTDESC>
        <REPORTNAME>All Masters</REPORTNAME>
      </REQUESTDESC>
      <REQUESTDATA>
        <TALLYMESSAGE xmlns:UDF=”TallyUDF”>
            <LEDGER Action=”Create”>
              <NAME>Customer ABC</NAME>
              <PARENT>Sundry Debtors</PARENT>
            </LEDGER>
            </TALLYMESSAGE>
        </REQUESTDATA>
</IMPORTDATA>
</BODY>
</ENVELOPE>
```

Adding Address Information in Ledger Master

The below mentioned XML tags can be included in the XML request to add the address information in ledger master.

```
<LEDGER Action=”Create”>
  <NAME>Customer ABC</NAME>
  <PARENT>Sundry Debtors</PARENT>
  <MAILINGNAME.LIST TYPE=”String”>
     <MAILINGNAME>Customer – Mailing name</MAILINGNAME>
  </MAILINGNAME.LIST>
  <ADDRESS.LIST TYPE=”String”>
     <ADDRESS>Door No </ADDRESS>
     <ADDRESS>Lane</ADDRESS>
     <ADDRESS>Locality</ADDRESS>
  </ADDRESS.LIST>
  <PINCODE>560068</PINCODE>
  <COUNTRYNAME>India</COUNTRYNAME>
  <LEDSTATENAME>Karnataka</LEDSTATENAME>
  <EMAIL>A@abc.com</EMAIL>
  <EMAILCC>ACC@abc.com</EMAILCC>
  <LEDGERPHONE>0888888</LEDGERPHONE>
  <LEDGERMOBILE>99999999</LEDGERMOBILE>
</LEDGER>
```

Similarly, we can create any ledger in Tally by specifying the appropriate group/parent.

Altering a Ledger

An existing ledger in Tally can be altered by sending a XML request. Only the tag <Ledger> has to be changed as mentioned below in the XML request demonstrated for creating of ledger.

```
<Ledger NAME=”Customer ABC” Action=”Alter”>
```

Guidelines

The tool Tally Connector available in TallyPrime Developer can be used for sending XML request and receiving the XML response for the quick understanding

Ensure the dependent masters are available in Tally before sending XML request for creating the master/ transactions. Even if the masters are available in Tally ensure that it is created appropriately under nature of group for example, Sales ledger should be available under group Sales Accounts, Purchase ledger should be created under group Purchase Accounts.

- Third party applications can ensure this by sending XML request and show the appropriate warning/error message to the user.
- Tally maintains a unique number (master-ID) for all the masters and transactions. Tally master-ID can be associated with Third party applications for additional validation, if required.
- It is advisable to create one nature of master at once to have better control, for example, Set of Groups and Set of ledgers needs to be created in two XML requests.
- The tax related information has to be sent to Tally to maintain the statutory records in Tally.

Inventory Masters

The Inventory masters are similar to Accounting Masters. Inventory masters are needed to maintain the stock details of an organization to realize the availability of stock in hand which is part of the Balance sheet.

The following table describes the type of inventory masters and its purpose:

Master	Information
Stock Group	Stock groups holds collection of stock items.
Stock Items	A Stock item is the actual inventory master used in the transactions
Unit of Measure	Stock Items are transacted based on the quantity. The quantity is measured by the Unit of Measure.
Locations/Godowns	To maintain the availability of stock at various locations. Stock items are stored under the Godowns

Stock Group

Stock Group in Inventory are similar to Groups in Accounting masters. Stock Groups are useful to classify the Stock Items. The stock group classification can be made based on some common features such as brand name, product type, quality, etc. Grouping helps to locate Stock Items easily and report their details in statements. The stock group can have collection of stock groups (sub stock groups) as well.

The following XML request can be send to Tally from a third party applications to create a stock group ‘Electronics’

```
<ENVELOPE>
<HEADER>
    <TALLYREQUEST>Import Data</TALLYREQUEST>
</HEADER>
<BODY>
<IMPORTDATA>
      <REQUESTDESC>
        <REPORTNAME>All Masters</REPORTNAME>
      </REQUESTDESC>
      <REQUESTDATA>
        <TALLYMESSAGE xmlns:UDF=”TallyUDF”>
            <STOCKGROUP Action=”Create”>
              <NAME>Electronics</NAME>
            </STOCKGROUP>
          </TALLYMESSAGE>
      </REQUESTDATA>
</IMPORTDATA>
</BODY>
</ENVELOPE>
```

Stock Item

Stock Item refers to the goods that an organization manufacture, trade or maintains. A stock item can be created by providing a Name in Tally. However, the inventory master UOM (Unit of Measure) is mandatory to define the quantity. It is good to create the dependent masters of the stock items before creating a stock item.

The following are the XML tags for creating a stock item:

Tag	Data type	Is mandatory	Description

NAME	String	Yes	Name of the stock item
BASEUNITS	String	No	Name of the UOM

Creating a Stock Item

Stock item can be created in Tally from a third party application by using the below mentioned sample XML request

```
<ENVELOPE>
     <HEADER>
        <TALLYREQUEST>Import data</TALLYREQUEST>
        <TYPE>Data</TYPE>
        <ID>All Masters</ID>
      </HEADER>
    <BODY>
        <IMPORTDATA>
            <REQUESTDESC>
                <REPORTNAME>All Masters</REPORTNAME>
            </REQUESTDESC>
            <REQUESTDATA>
                <TALLYMESSAGE xmlns:UDF=”TallyUDF”>
                    <STOCKITEM Action=”Create”>
                       <NAME>Red Colored Striped Shirt</NAME>
                        <BASEUNITS>lts</BASEUNITS>
                        <NAME.LIST TYPE=”String”>
                          <NAME>Red Colored Striped Shirt</NAME>
                        </NAME.LIST>
                    </STOCKITEM>
                </TALLYMESSAGE>
            </REQUESTDATA>
        </IMPORTDATA>
    </BODY>
```

```
</ENVELOPE>
```

NOTE To create multiple stock items, the tag set **<Stockitem>** can be repeated as required.

Altering of a Stock Item

You can do the alteration of stock item using the following tag and value

```
<STOCKITEM NAME=”Red Colored Striped Shirt” Action=”Alter”>
```

NOTE UOM of a stock item cannot be altered when the stock item used in a transaction or in a master.

XML Response for Import Data Request

Tally responses in the following XML format for the XML request:

Tag	Description
<RESPONSE>	Opening tag of response XML
<CREATED>0</CREATED>	Number of masters/transactions created
<ALTERED>0</ALTERED>	Number of masters/transactions altered
<DELETED>0</DELETED>	Number of masters/transactions deleted
<LASTVCHID>0</LASTVCHID>	Master ID of last imported voucher
<LASTMID>0</LASTMID>	Last Master ID (it always returns 0)
<COMBINED>0</COMBINED>	Number of combined masters
<IGNORED>0</IGNORED>	Number of ignored masters
<ERRORS>0</ERRORS>	Number of errors while importing the request
<CANCELLED>0</CANCELLED>	Number of transactions cancelled
</RESPONSE>	Closing tag of response XML

Other Optional Attributes of a Stock Item

Depending on the features enabled in Tally, the tags can be added in the XML request.

Example:

Name	Usage of the Tag	Tag
Standard Selling price	To provide standard selling for stock item	<STANDARDPRICELIST.LIST> <DATE>14092016</DATE> <RATE>1145.99</RATE> </STANDARDPRICELIST.LIST>
Standard Cost price	To provide standard cost for stock item	<STANDARDCOSTLIST.LIST> <DATE>12092016</DATE> <RATE>875.23</RATE> </STANDARDCOSTLIST.LIST>
Parent	To provide stock group name to stock item name	<PARENT> "Electronics" </PARENT>
BatchName	To provide batch name	<BATCHNAME> Apr2016 </BATCHNAME>
Opening balance	To provide opening quantity for the item	<OPENINGBALANCE>25 </OPENINGBALANCE>
Opening Rate	To provide opening rate for the item	<OPENINGRATE>490 </OPENINGRATE>
Opening Value	To provide opening value for the item	<OPENINGVALUE>12250 </OPENINGVALUE>

Godown Name	To provide godown name. Incase, godown feature is not enabled, the name of godown can be set to Main Location as this is the default value in Tally	<GODOWN> Main Location </GODOWN>

Identify/Map Items Pushed by Third Party Applications

There may be a need to map the stock items maintained in Tally Vs stock items maintained in third party applications. In such case, it is necessary to store a reference value in stock item master. This can be achieved using the fields Alias or Part number of the stock item. Tally has a capability to provide multiple Aliases and Part numbers to a master.

Alias

The **Alias** field gets enabled by the option “Provide aliases along with name?” in stock item master configuration.

Tag	Data type	Is mandatory	Description
NAME.LIST	List	Yes	It is the header of alias list
NAME	String	Yes	Alias name

Third party applications can store multiple unique aliases using this following XML tags.

```
<NAME.LIST TYPE=”String”>
    <NAME>Red Striped Shirt</NAME>
    <NAME>Stockno12345</NAME>
    <NAME>TPA-Item-001</NAME>
```

```
</NAME.LIST>
```

Part No

The **Part No.** field gets enabled by the option "**Use Part Number for stock items?**" in Stock item master configuration. This field is used to enter the catalogue number.

Tag	Data type	Is mandatory	Description
MAILINGNAME.LIST	List	Yes	It is the header of part number list
MAILINGNAME	String	Yes	To provide the Part number

Third party applications can store multiple unique Part numbers using this following XML tags.

```
<MAILINGNAME.LIST TYPE="String">
    <MAILINGNAME>TPA-Stockno-00001</MAILINGNAME>
    <MAILINGNAME>00001</MAILINGNAME>
</MAILINGNAME.LIST>
```

Units of Measure

Stock Items are purchased or sold on the basis of quantity. The quantity is measured by Units. There are two types of UOMs: Simple and Compound. Simple unit is for basic requirement such as numbers, meters, kilograms, and pieces. Compound units can be used with the combination of two UOMs, like 1 box of 10 pieces [1 box = 10 pieces], to measure the stock items in two aspects

Simple UOM

Simple units are nos, pcs, and so on.

Compound UOM

A **Compound unit** is a relation between two Simple Units. Hence, before you create a Compound Unit, ensure that you have already created two Simple Units.

For example, To Create Compound unit – Doz (Dozen) of 12 Nos (Numbers), you have to create two simple units, Doz (Dozen) and Nos (Numbers) and set the conversion factor as 12.

UOM contains the following fields.

Tag	Data type	Is mandatory	Description

NAME	String	Yes	Name of the UOM. This would be used as symbol for unit. (Applicable for simple unit. however, Tally creates the name for compound unit based on the conversion factor of Base unit and additional unit)
ISSIMPLEUNIT	Logical	Yes	To set the unit as simple or compound
ORIGINALNAME	String	No	Formal name for the unit
DECIMALPLACES	Number	No	To provide number of decimal places. (Applicable for simple unit)
BASEUNITS	String	Yes	To provide the name of base unit (Applicable for Compound unit)
ADDITIONALUNITS	String	Yes	To provide the name of additional unit (Applicable for Compound unit)
CONVERSION	Number	Yes	To provide the conversion factor from base to additional unit (Applicable for Compound unit)

Simple & Compound Unit Creation

XML tags for simple unit and compound unit creation are as given below:

NOTE In compound unit creation, <**NAME**> tag value will not be considered. Tally creates the name for compound unit based on the conversion factor of Base unit and additional unit.

Simple Unit	Compound Unit
<ENVELOPE> <HEADER> <TALLYREQUEST>Import Data</TALLYREQUEST> </HEADER> <BODY> <IMPORTDATA> <REQUESTDESC> <REPORTNAME>All Masters </REPORTNAME> </REQUESTDESC> <REQUESTDATA> <TALLYMESSAGE xmlns:UDF=”TallyUDF”> <UNIT Action = “Create”> <NAME>Pcs</NAME> <ISSIMPLEUNIT>Yes</ISSIMPLEUNIT> <ORIGINALNAME>Pieces </ORIGINALNAME> <DECIMALPLACES>2</DECIMALPLACES> </UNIT> <UNIT Action = “Create”> <NAME>Doz</NAME> <ISSIMPLEUNIT>Yes</ISSIMPLEUNIT> <ORIGINALNAME>Dozen </ORIGINALNAME> <DECIMALPLACES>0</DECIMALPLACES> </UNIT>	<ENVELOPE> <HEADER> <TALLYREQUEST>Import Data</TALLYREQUEST> </HEADER> <BODY> <IMPORTDATA> <REQUESTDESC> <REPORTNAME>All Masters </REPORTNAME> </REQUESTDESC> <REQUESTDATA> <TALLYMESSAGE xmlns:UDF=”TallyUDF”> <UNIT Action = “Create”> <NAME>Dozen of 12 Pieces</NAME> <ISSIMPLEUNIT>No</ISSIMPLEUNIT> <ORIGINALNAME>Liters</ORIGINALNAME> <DECIMALPLACES>2</DECIMALPLACES> <BASEUNITS>Doz</BASEUNITS> <ADDITIONALUNITS>Pcs</ADDITIONALUNITS> <CONVERSION>12</CONVERSION> </UNIT> </TALLYMESSAGE> </REQUESTDATA> </IMPORTDATA> </BODY> </ENVELOPE>

```
            </TALLYMESSAGE>

             </REQUESTDATA>

        </IMPORTDATA>

        </BODY>

        </ENVELOPE>
```

Location/Godown

Location/Godown is the place where Stock Items are stored. You can obtain stock reports for each godown and account for the movement of stock between locations/godowns. Click here for the steps to create a Godown, manually in Tally.

The following XML request can be send to Tally from a third party applications to create a Location/Godown ‘Factory’

```
<ENVELOPE>

<HEADER>

    <TALLYREQUEST>Import Data</TALLYREQUEST>

</HEADER>

<BODY>

<IMPORTDATA>

      <REQUESTDESC>

        <REPORTNAME>All Masters</REPORTNAME>

      </REQUESTDESC>

      <REQUESTDATA>

        <TALLYMESSAGE xmlns:UDF=”TallyUDF”>

            <GODOWN Action=”Create”>

              <NAME>Factory</NAME>

            </GODOWN>

        </TALLYMESSAGE>

      </REQUESTDATA>

</IMPORTDATA>

</BODY>

</ENVELOPE>
```

NOTE There is a default godown ‘Main Location’ available in Tally. Incase the feature ‘Maintain multiple godowns?’ is not enabled in Tally, the name of godown can be set to ‘Main Location’, as it is a default value in masters and transactions

Guidelines

The mandatory tags required to import/create the inventory masters in Tally is explained in this document. To understand other tags, create a sample master with the required features in Tally and export the same in XML format. Click here for the steps to export the masters from Tally.

Extracting/pulling the Inventory masters from Tally to third party applications is covered under the section Export Reports/Data from Tally.

Transactions

The **Voucher** is the document used for recording all type of transactions in accounting standards. It can be either impact of cash or material between two entities. There are various voucher types available in Tally to categorize the transactions as Purchase, Sales, Payment, Receipt, Journal etc. The details of all voucher types are available here.

Sales Transaction

The **Sales voucher** is considered for explaining the integration of sales vouchers from third party applications to Tally.

Sales voucher type used to record the sales transactions of an organization. A bill is generated on sale of goods or services provided or both sales and services. All these requirements fulfilled in Tally. There are two modes, ‘As voucher’ and ‘As invoice’, available in Tally to record the sales transactions. Both the mode supports to enter the transactions with or without inventory details. The mode ‘As Invoice’ prints the voucher in Invoice format and it can be hand it over to the Buyer / Party for proof of transaction.

- The mandatory XML tags for a Sale transaction is given below. However, it has to specified depends upon the mode of transaction. The structure and hierarchy of each mode explained below with sample XML requests.

Tags	Data Type	Permissible Values	Description
VOUCHER VCHTYPE=”VOUCHERTYPENAME” ACTION=”Create” OBJVIEW=”Accounting Voucher View”	Static Values	VCHTYPE is to provide voucher type name ACTION is to provide action name OBJVIEW is to	This is opening tag for voucher where the OBJVIEW parameter decides the type of voucher, vchtype informs about

		provide the object view	voucher type and action undertakes the action to be performed on object voucher
PERSISTEDVIEW	String	Accounting Voucher View Invoice Voucher View Inventory Voucher View Pay Slip Voucher View Consumption Voucher View Multi Consumption Voucher View	To identify the Voucher behaviour.
VOUCHERTYPENAME	String	Name of the voucher type	To provide voucher type name
DATE	Date	Uni date in format of yyyymmdd	To provide date for the voucher
ISINVOICE	Logical	Boolean	To identify whether the voucher should be recorded as invoice or voucher.
ALLLEDGERENTRIES.LIST	List Tag		To provide ledger details of the voucher
LEDGERNAME	String	Ledger Name	To provide ledger name
ISDEEMEDPOSITIVE	Logical	Boolean	This tag decides whether the ledger amount to be debited or credited. In sales

			voucher this should be set to YES for all debiting a ledger
ISPARTYLEDGER	Logical	Boolean	To identify whether the ledger is a party ledger or not
AMOUNT	Amount	Amount	To provide amount for the ledger
BILLALLOCATIONS.LIST	List Tag		This is opening tag for providing bill details for party ledger and is mandate only for party ledger. For other ledgers the tag is not required
NAME	String	Text	To provide name / number for the bill
BILLTYPE	String	Advance Agst Ref New Ref On Account	To identify the payment type received from party. This would impact bill outstanding report
AMOUNT	Amount	Amount	To provide bill amount
/BILLALLOCATIONS.LIST	List Tag		This is closing tag of bill allocation
/ALLLEDGERENTRIES.LIST	List Tag		This is closing tag for ledger details
ALLINVENTORYENTRIES.LIST	List Tag		This is opening tag for Inventory Entries of voucher

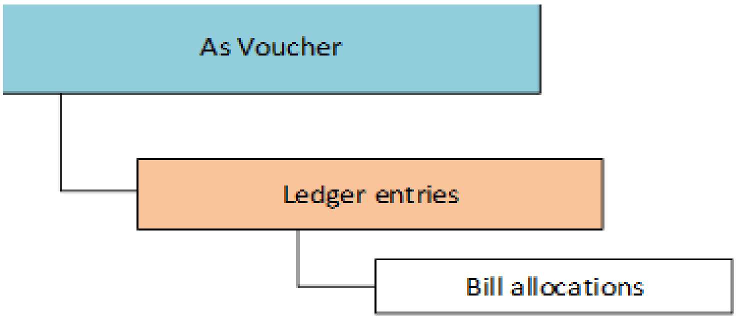
STOCKITEMNAME	String		To provide stock item name
ISDEEMEDPOSITIVE	Logical	Boolean	This tag is set to NO for outward transactions
ACTUALQTY	Quantity	Number	To provide actual quantity of the item
BILLEDQTY	Quantity	Number	To provide billed quantity of the item
RATE	Rate	Number	To provide rate for billed quantity
AMOUNT	Amount		To provide amount for the item
ACCOUNTINGALLOCATIONS.LIST	List Tag		This is the opening tag to provide accounting details for item
LEDGERNAME	String	Text	To provide sales / Tax ledger name
ISDEEMEDPOSITIVE	Logical	Boolean	This tag decides whether the ledger amount to be debited or credited. In sales voucher this should be set to No for Sales ledger
AMOUNT	Number	Amount	To provide amount for the ledger
/ACCOUNTINGALLOCATIONS.LIST	List Tag		This is a closing

			tag for accounting allocations
BATCHALLOCATIONS.LIST	List Tag	#NA	This is opening tag for Batch allocations
GODOWNNAME	String		To provide godown name. Incase godown is not enabled but batch is used the value has to be Main Location
BATCHNAME	String		To provide batch name. Incase batch is not enabled but godown / tracking number is used the value has to be Primary Batch
AMOUNT	Amount		This tag is to provide amount for batch
ACTUALQTY	Quantity		This tag is to provide actual quantity sent by the supplier. This would affect all inventory reports
BILLEDQTY	Quantity		To provide the quantity billed
/BATCHALLOCATIONS.LIST	List Tag		This is a closing tag for Batch allocations
/ALLINVENTORYENTRIES.LIST	List Tag		This is a closing tag for Inventory

			Entries of voucher
/VOUCHER	List Tag		This is a closing tag for voucher

Sales Transaction – As Voucher

The voucher mode is the traditional way to record an entry with only the accounting information, i.e., Recording the entry in Dr-Cr or By-To format to specify ledger amount either debit and credit.



Sample XML Request for Creating a sales transaction in voucher mode

```
<ENVELOPE>
<HEADER>
  <VERSION>1</VERSION>
  <TALLYREQUEST>Import</TALLYREQUEST>
  <TYPE>Data</TYPE>
  <ID>Vouchers</ID>
</HEADER>
<BODY>
<DESC>
</DESC>
<DATA>
<TALLYMESSAGE>
<VOUCHER>
   <DATE>20160401</DATE>
   <VOUCHERTYPENAME>Sales</VOUCHERTYPENAME>
   <VOUCHERNUMBER>1</VOUCHERNUMBER>
   <PERSISTEDVIEW>Accounting Voucher View</PERSISTEDVIEW>
   <ISINVOICE>No</ISINVOICE>
  <LEDGERENTRIES.LIST>
   <LEDGERNAME>ABC Company Limited</LEDGERNAME>
   <ISDEEMEDPOSITIVE>Yes</ISDEEMEDPOSITIVE>
   <ISPARTYLEDGER>Yes</ISPARTYLEDGER>
   <ISLASTDEEMEDPOSITIVE>Yes</ISLASTDEEMEDPOSITIVE>
   <AMOUNT>-215476.00</AMOUNT>
  </LEDGERENTRIES.LIST>
  <LEDGERENTRIES.LIST>
     <LEDGERNAME>Sales</LEDGERNAME>
     <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
     <AMOUNT>21546.00</AMOUNT>
   </LEDGERENTRIES.LIST>
</VOUCHER>
</TALLYMESSAGE>
</DATA>
</BODY>
</ENVELOPE>
```

Note 1: The inventory details can be added by just adding the below tags under sales ledger as the inventory details are the break-up of sales ledger.

```
 <INVENTORYALLOCATIONS.LIST>
    <STOCKITEMNAME>SAMSUNG DVD PLAYER</STOCKITEMNAME>
    <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
    <AMOUNT>21546.00</AMOUNT>
    <ACTUALQTY> 1 nos</ACTUALQTY>
    <BILLEDQTY> 1 nos</BILLEDQTY>
    <BATCHALLOCATIONS.LIST>
     <GODOWNNAME>Factory</GODOWNNAME>
     <BATCHNAME>Primary Batch</BATCHNAME>
     <AMOUNT>21546.00</AMOUNT>
     <ACTUALQTY> 1 nos</ACTUALQTY>
     <BILLEDQTY> 1 nos</BILLEDQTY>
```

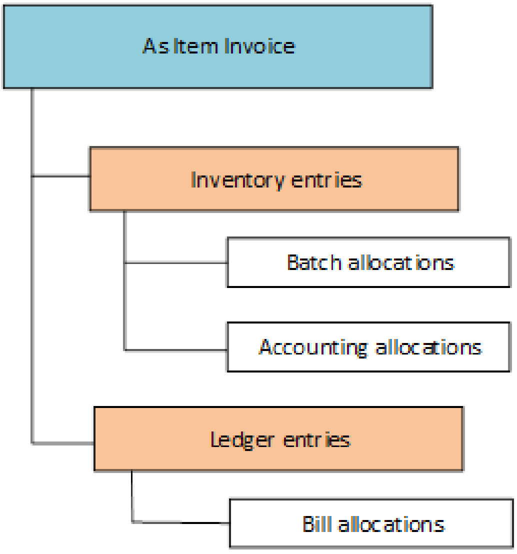
```
        </BATCHALLOCATIONS.LIST>
       </INVENTORYALLOCATIONS.LIST>
```

Note 2: As stated earlier, an invoice can be recorded without inventory entries for the purpose of preparing service bills by changing below mentioned two tags. Such transactions called as ‘Accounting Invoice’ in Tally.

```
<PERSISTEDVIEW>Invoice Voucher View</PERSISTEDVIEW>
<ISINVOICE>Yes</ISINVOICE>
```

Sales Transaction – As Invoice

The Item invoice recorded in Tally using the Invoice mode where the calculations can be automated and the user can experience the ease of feeding the transactions into the system. The invoice mode is the modern way to record an entry. i.e. you simply provide required data and an auto calculation would take place.



Voucher Creation

This below mentioned XML request imports a voucher, As Item Invoice, in TallyPrime. The same can be viewed in the reports where the sales transactions are captured, For e.g. Day Book, Sales Register, etc.

Sample XML Request for Creating a Sales Transaction in Invoice Mode

```
<ENVELOPE>
<HEADER>
   <VERSION>1</VERSION>
   <TALLYREQUEST>Import</TALLYREQUEST>
   <TYPE>Data</TYPE>
   <ID>Vouchers</ID>
</HEADER>
```

```
<BODY>
<DESC></DESC>
<DATA>
<TALLYMESSAGE>
<VOUCHER>
 <DATE>20160401</DATE>
 <VOUCHERTYPENAME>Sales</VOUCHERTYPENAME>
 <VOUCHERNUMBER>1</VOUCHERNUMBER>
 <PERSISTEDVIEW>Invoice Voucher View</PERSISTEDVIEW>
 <ISINVOICE>Yes</ISINVOICE>
 <OBJVIEW>Invoice Voucher View</OBJVIEW>
 <LEDGERENTRIES.LIST>
  <LEDGERNAME>ABC Company Limited</LEDGERNAME>
  <ISDEEMEDPOSITIVE>Yes</ISDEEMEDPOSITIVE>
  <ISPARTYLEDGER>Yes</ISPARTYLEDGER>
  <ISLASTDEEMEDPOSITIVE>Yes</ISLASTDEEMEDPOSITIVE>
  <AMOUNT>-21546.00</AMOUNT>
  <BILLALLOCATIONS.LIST>
   <NAME>1</NAME>
   <BILLTYPE>New Ref</BILLTYPE>
   <AMOUNT>-21546.00</AMOUNT>
  </BILLALLOCATIONS.LIST>
</LEDGERENTRIES.LIST>
<LEDGERENTRIES.LIST>
  <LEDGERNAME>Packing Charges</LEDGERNAME>
  <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
  <AMOUNT>900.00</AMOUNT>
</LEDGERENTRIES.LIST>
<LEDGERENTRIES.LIST>
  <BASICRATEOFINVOICETAX.LIST TYPE="Number">
  <BASICRATEOFINVOICETAX> 14</BASICRATEOFINVOICETAX>
  </BASICRATEOFINVOICETAX.LIST>
```

```
      <ROUNDTYPE/>
      <LEDGERNAME>VAT</LEDGERNAME>
      <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
      <AMOUNT>2646.00</AMOUNT>
    </LEDGERENTRIES.LIST>
    <ALLINVENTORYENTRIES.LIST>
      <STOCKITEMNAME>Sony Television 14 inches</STOCKITEMNAME>
      <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
      <RATE>15000.00/nos</RATE>
      <AMOUNT>15000.00</AMOUNT>
      <ACTUALQTY> 1 nos</ACTUALQTY>
      <BILLEDQTY> 1 nos</BILLEDQTY>
       <BATCHALLOCATIONS.LIST>
          <GODOWNNAME>Main Location</GODOWNNAME>
          <BATCHNAME>Primary Batch</BATCHNAME>
          <DESTINATIONGODOWNNAME>Main Location</DESTINATIONGODOWNNAME>
          <AMOUNT>15000.00</AMOUNT>
          <ACTUALQTY> 1 nos</ACTUALQTY>
          <BILLEDQTY> 1 nos</BILLEDQTY>
        </BATCHALLOCATIONS.LIST>
        <ACCOUNTINGALLOCATIONS.LIST>
           <LEDGERNAME>Sales</LEDGERNAME>
           <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
           <AMOUNT>15000.00</AMOUNT>
        </ACCOUNTINGALLOCATIONS.LIST>
      </ALLINVENTORYENTRIES.LIST>
      <ALLINVENTORYENTRIES.LIST>
        <STOCKITEMNAME>Samsung DVD Player</STOCKITEMNAME>
        <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
        <RATE>3000.00/nos</RATE>
        <AMOUNT>3000.00</AMOUNT>
        <ACTUALQTY> 1 nos</ACTUALQTY>
      <BILLEDQTY> 1 nos</BILLEDQTY>
    <BATCHALLOCATIONS.LIST>
       <GODOWNNAME>Main Location</GODOWNNAME>
       <BATCHNAME>Primary Batch</BATCHNAME>
       <DESTINATIONGODOWNNAME>Main Location</DESTINATIONGODOWNNAME>
            <AMOUNT>3000.00</AMOUNT>
            <ACTUALQTY> 1 nos</ACTUALQTY>
            <BILLEDQTY> 1 nos</BILLEDQTY>
        </BATCHALLOCATIONS.LIST>
        <ACCOUNTINGALLOCATIONS.LIST>
             <LEDGERNAME>Sales</LEDGERNAME>
             <ISDEEMEDPOSITIVE>No</ISDEEMEDPOSITIVE>
             <AMOUNT>3000.00</AMOUNT>
         </ACCOUNTINGALLOCATIONS.LIST>
       </ALLINVENTORYENTRIES.LIST>
   </VOUCHER>
</TALLYMESSAGE>
</DATA>
</BODY>
</ENVELOPE>
```

XML Response

On processing the above request for importing vouchers, the requested vouchers are created in Tally and the returns the response as mentioned below:

The above XML Response is a log of vouchers created, altered, combined, ignored or not imported due to some errors. It also contains information pertaining to last Voucher ID imported.

Voucher Alteration

In case of voucher alteration or cancellation, a vital information required is voucher identifier. Identification of voucher can be achieved through Master ID, Voucher Number, Voucher Type and date

Sample XML Request for Voucher Alteration

```
<ENVELOPE>
 <HEADER>
    <VERSION>1</VERSION>
    <TALLYREQUEST>Import</TALLYREQUEST>
    <TYPE>Data</TYPE>

    <ID>Vouchers</ID>

 </HEADER>

    <BODY>

    <DESC></DESC>

    <DATA>

    <TALLYMESSAGE>

       <VOUCHER DATE=”01-Apr-2016" TAGNAME =”Voucher Number”

                TAGVALUE=”1" Action=”Alter” VCHTYPE = “Sales”>

       <NARRATION>Being Goods sold</NARRATION>

       </VOUCHER>

    </TALLYMESSAGE>

    </DATA>

    </BODY>

</ENVELOPE>
```

Voucher Cancellation

Voucher cancellation is similar to above Voucher Alteration. For Voucher Cancellation, Action must be set to “Cancel”

Sample XML Request for cancellation

```
<ENVELOPE>

<HEADER>

  <VERSION>1</VERSION>

  <TALLYREQUEST>Import</TALLYREQUEST>

  <TYPE>Data</TYPE>

  <ID>Vouchers</ID>

 </HEADER>

 <BODY>

   <DESC></DESC>

   <DATA>

   <TALLYMESSAGE>

   <VOUCHER DATE=”01-Apr-2016" TAGNAME = “VoucherNumber” TAGVALUE=”2"

    ACTION=”Cancel” VCHTYPE = “Sales”>
   <NARRATION>Being good returned</NARRATION>
  </VOUCHER>
  </TALLYMESSAGE>
</DATA>
</BODY>
</ENVELOPE>
```

Guidelines

To understand or import the values of other than mandatory tags, create the expected voucher entry in TallyPrime. and export the same in XML format. This is the format of the XML the third party applications must generate. So, pick up the relevant tags from here and place it as per the hierarchy in your XML requests.

Examples of few possible errors

DESC not found

Sample XML response:

```
<ENVELOPE>
 <HEADER>
   <VERSION>1</VERSION>
   <STATUS>0</STATUS>
  </HEADER>
  <BODY>
   <DATA>DESC not found </DATA>
  </BODY>
</ENVELOPE>
```

Solution: Ensure the specified masters in the XML are created and available in Tally.

Voucher totals do not match!

Sample XML response:

```
<ENVELOPE>
<HEADER>
  <VERSION>1</VERSION>
  <STATUS>0</STATUS>
 </HEADER>
 <BODY>
<DATA>
<LINEERROR>Voucher totals do not match! Dr: 20,394.00 Dr Cr: 20,395.00 Cr Diff: 1.00 Cr </LINEERROR>
 <CREATED>0</CREATED>
 <ALTERED>0</ALTERED>
 <DELETED>0</DELETED>
 <LASTVCHID>0</LASTVCHID>
 <LASTMID>0</LASTMID>
 <COMBINED>0</COMBINED>
 <IGNORED>0</IGNORED>
 <ERRORS>1</ERRORS>
 <CANCELLED>0</CANCELLED>
 <VCHNUMBER>1</VCHNUMBER>
 <DESC></DESC>
 </DATA>
</BODY>
</ENVELOPE>
```

Solution: Ensure the total of debit amounts are equals to the total of credit amounts

> NOTE Apart from the XML response, the statistics of import execution logs the file Tally.imp as well, under the folder where the TallyPrime is installed.

Prerequisites to Import Vouchers

- Make sure that the necessary masters (Ledger, Stock Item, UOM, etc.) exists in TallyPrime
- In case non-base currencies are used, make sure these currencies are available in TallyPrime
- The totals of Debit values and Credit values of Voucher should be equal
- All dates must follow the YYYYMMDD format.

Reports

This section is intended to illustrate the approaches to get the data and reports from Tally through XML request/ response.

Tally Data/Objects

Tally database is a true OODBMS(Object Oriented Database Management System). It is possible to store data as objects and retrieve data as objects. In other traditional OODBMS, data pertaining to an object is internally converted into disparate files and stored, but in Tally database data for an object is stored as it is as a block. This allows faster retrieval of data.

Tally File System consists of data files (master, transaction, link masters), Msgfiles (transaction management) and State Files (concurrency control and exclusivity control). It follows the concept of embedded and weighted Indexes. In other traditional databases, separate index file is maintained. In Tally, indexes are built into the data files. This facilitates faster retrieval of data.

By design, Tally database is hierarchical in nature i.e., objects are stored in a tree like structure. Each node in the tree can be a tree in itself. In this structure, a parent can have multiple children however every child can have only one parent. A child can further have multiple children. All the characteristics of a child are inherited from its parent and no child can exist without a parent.

Reports

Tally creates the books of accounts and the financial statements based on the vouchers entered for the particular period. The information is designed to allow a user to get the maximum benefit of the data that is entered. A user gets a holistic picture of the data and he able to present information using different options. The purpose of compiling data is to present it in comprehensible accounting reports. On entering the vouchers, TallyPrime uses the same data and provides you with the management control intelligences in addition to all books and statements.

The reports of Tally can be gathered through XML request/response.

Tags used for sending a request to export data from TallyPrime as follows.

<HEADER> contains the following:

- Tag <TALLYREQUEST> must contain value **Export**
- Tag <TYPE> must contain value **Data**
- Tag <ID> should contain the **Name of the Report**

<BODY> contains the following:

- Tag <DESC> can contain report configurations like Company Name, Format of export, etc. as desired and which should be enclosed within <STATICVARIABLES> tag list.
- If the Report Name specified in the <ID> tag does not exist within Tally running at the specified port, the TDL defining the Report & other supporting definition needs to be described and enclosed within tag <TDL>.

Gathering data and reports from Tally

Gathering data

The data of any master or transaction in Tally can be pulled through XML request/response. There are three ‘type’ of requests used for exporting the data from Tally and the purpose of each type is given below:

Object	To get one particular object. For example, Ledger, Stock Item

Collection	To get multiple objects. For example, List of Ledgers, List of Stockitems
Data	To get the reports. For example, Balance Sheet, Trial Balance

The below mentioned XML request gets the Name and Parent of a single ledger from Tally

```
<ENVELOPE>
<HEADER>
    <VERSION>1</VERSION>
    <TALLYREQUEST>EXPORT</TALLYREQUEST>
    <TYPE>OBJECT</TYPE>
    <SUBTYPE>Ledger</SUBTYPE>
    <ID TYPE=”Name”>NameofTheLedger</ID>
</HEADER>
<BODY>
<DESC>
    <STATICVARIABLES>
       <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>
    </STATICVARIABLES>
    <FETCHLIST>
       <FETCH>Name</FETCH>
       <FETCH>Parent</FETCH>
    </FETCHLIST>
</DESC>
</BODY>
</ENVELOPE>
```

The below mentioned XML request gets the data of all stock items from Tally

```
<ENVELOPE>
<HEADER>
    <VERSION>1</VERSION>
    <TALLYREQUEST>Export</TALLYREQUEST>
    <TYPE>Data</TYPE>

    <ID>List of Accounts</ID>

</HEADER>

<BODY>

<DESC>

<STATICVARIABLES>

      <AccountType>Stock Items</AccountType>

</STATICVARIABLES>

</DESC>

</BODY>

</ENVELOPE>
```

The below mentioned XML request gets the data of ledgers of a particular group ‘Bank accounts’.

> NOTE The default report ‘List of Accounts’ in Tally provides all the masters created in a company. Hence, an XML request can be sent to Tally to get details for the required masters like Ledgers, Groups, Stock Items, Stock Groups, etc.

The below mentioned XML request gets the data of ledgers of a particular group ‘Bank accounts’

```
<ENVELOPE>

<HEADER>

    <VERSION>1</VERSION>

    <TALLYREQUEST>EXPORT</TALLYREQUEST>

    <TYPE>COLLECTION</TYPE>

    <ID>List of Ledgers</ID>

</HEADER>

<BODY>

<DESC>

<STATICVARIABLES>

      <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>

</STATICVARIABLES>

<TDL>

<TDLMESSAGE>

    <COLLECTION NAME=”List of Ledgers” ISMODIFY=”Yes”>

      <ADD>CHILD OF : Bank Accounts</ADD>
      <NATIVEMETHOD>Name</NATIVEMETHOD>

      <NATIVEMETHOD>Parent</NATIVEMETHOD>

    </COLLECTION>

</TDLMESSAGE>

</TDL>

</DESC>

</BODY>

</ENVELOPE>
```

NOTE The default collection ‘List of Ledgers’ is available in Tally and it is altered in the above request to pull only the ledgers created under the group ‘Bank Accounts’

Similar to masters, the data of transactions can be gathered from Tally through XML request. The below mentioned XML request gets data of transactions for a specific period.

```
<ENVELOPE>

<HEADER>

      <VERSION>1</VERSION>

      <TALLYREQUEST>Export</TALLYREQUEST>

      <TYPE>Data</TYPE>

      <ID>DayBook</ID>

</HEADER>

<BODY>

<DESC>

  <STATICVARIABLES>

    <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>

        <SVFROMDATE TYPE=”Date”>1-Apr-2016</SVFROMDATE>

        <SVTODATE TYPE=”Date”>1-Apr-2016</SVTODATE>

  </STATICVARIABLES>

</DESC>

</BODY>

</ENVELOPE>
```

The below XML request gets data of transactions for a specific period and for specific voucher type

```
<ENVELOPE>

 <HEADER>
```

```
      <VERSION>1</VERSION>

      <TALLYREQUEST>Export</TALLYREQUEST>

      <TYPE>Data</TYPE>

      <ID>Daybook</ID>

 </HEADER>

<BODY>

<DESC>

<STATICVARIABLES>

    <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>

</STATICVARIABLES>

<TDL>

<TDLMESSAGE>

    <REPORT NAME=”Day Book” ISMODIFY=”Yes” ISFIXED=”No” ISINITIALIZE=”No” ISOPTION=”No”
    ISINTERNAL=”No”>

    <LOCAL>Collection : Default : Add :Filter : VchTypeFilter</LOCAL>

    <LOCAL>Collection : Default : Add :Fetch  : VoucherTypeName</LOCAL> </REPORT>

    <SYSTEM TYPE=”Formulae” NAME=”VchTypeFilter” ISMODIFY=”No” ISFIXED=”No”
    ISINTERNAL=”No”>$VoucherTypeName=Sales   </SYSTEM>

</TDLMESSAGE>

</TDL>

</DESC>

</BODY>

</ENVELOPE>
```

NOTE The default report ‘Day book’ in Tally provides all the transactions entered in a company. Hence, an XML request can be sent to Tally to get the details of the required vouchers. In the above example, the vouchers are filtered for a specific voucher type by specifying a formula using TDL.

Gathering Reports

The reports available in Tally can be gathered through XML request/response. The below mentioned XML request gets the existing report ‘Trial Balance’ from Tally.

```
<ENVELOPE>

<HEADER>
```

```
  <VERSION>1</VERSION>

  <TALLYREQUEST>Export</TALLYREQUEST>

  <TYPE>Data</TYPE>

  <ID>Trial Balance</ID>

</HEADER>

<BODY>

<DESC>

<STATICVARIABLES>

   <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>

</STATICVARIABLES>

</DESC>

</BODY>

</ENVELOPE>
```

Sample XML response for above request:

```
<ENVELOPE>

<DSPACCNAME>

  <DSPDISPNAME>Current Liabilities</DSPDISPNAME>

</DSPACCNAME>

<DSPACCINFO>

     <DSPCLDRAMT>

     <DSPCLDRAMTA>

     </DSPCLDRAMTA>

     </DSPCLDRAMT>

  <DSPCLCRAMT>

  <DSPCLCRAMTA>1526292.00</DSPCLCRAMTA>

  </DSPCLCRAMT>

  </DSPACCINFO>

  <DSPACCNAME>

      <DSPDISPNAME>Current Assets</DSPDISPNAME>

  </DSPACCNAME>

  <DSPACCINFO>

  <DSPCLDRAMT>
```

```
    <DSPCLDRAMTA>-43092.00</DSPCLDRAMTA>

  </DSPCLDRAMT>

  <DSPCLCRAMT>

    <DSPCLCRAMTA></DSPCLCRAMTA>

  </DSPCLCRAMT>

  </DSPACCINFO>

  <DSPACCNAME>

    <DSPDISPNAME>Sales Accounts</DSPDISPNAME>

  </DSPACCNAME>

  <DSPACCINFO>

  <DSPCLDRAMT>

     <DSPCLDRAMTA></DSPCLDRAMTA>

  </DSPCLDRAMT>

  <DSPCLCRAMT>

     <DSPCLCRAMTA>36000.00</DSPCLCRAMTA>

  </DSPCLCRAMT>

 </DSPACCINFO>

  <DSPACCNAME>

     <DSPDISPNAME>Purchase Accounts</DSPDISPNAME>

  </DSPACCNAME>

  <DSPACCINFO>

  <DSPCLDRAMT>

     <DSPCLDRAMTA>-1521000.00</DSPCLDRAMTA>

  </DSPCLDRAMT>

  <DSPCLCRAMT>

     <DSPCLCRAMTA></DSPCLCRAMTA>

  </DSPCLCRAMT>

  </DSPACCINFO>

  <DSPACCNAME>

  <DSPDISPNAME>Indirect Expenses</DSPDISPNAME>

  </DSPACCNAME>

  <DSPACCINFO>
```

```
    <DSPCLDRAMT>
        <DSPCLDRAMTA></DSPCLDRAMTA>
    </DSPCLDRAMT>
    <DSPCLCRAMT>
        <DSPCLCRAMTA>1800.00</DSPCLCRAMTA>
    </DSPCLCRAMT>
</DSPACCINFO>
</ENVELOPE>
```

In the above XML request, <HEADER> describes the expected result.

- The value of the Tag <TALLYREQUEST> is **Export**, which indicates that some information needs to be exported from Tally.
- The value of the Tag <TYPE> is **Data**, which indicates that the data needs to be exported from Tally. The value of the Tag <ID> must be a **TDL Report Name**, if the previous Tag <TYPE> contains Data and Tag <TALLYREQUEST> contains Export. Any **Report**, which needs to be exported from Tally, needs to be specified within these Tags.
- <BODY> Tag contains parameters, if any. Additional settings for the report like format required, company from which data is required, etc. can be passed within <STATICVARIABLES> Tag enclosed within <DESC> Tag. All variables are considered as Tag Names and their value are enclosed within these tags.

For example, in the above sample XML, Variable **SVEXPORTFORMAT** is considered as Tag and its value $$SysName:XML is enclosed within. TDL Internal Function SysName is evaluated at Tally end and the response is being sent accordingly.

Example:

```
<STATICVARIABLES>
    <SVCurrentCompany>ABC company Private Limited</SVCurrentCompany>
    <SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>
    <SVFROMDATE TYPE=”Date”>1-Apr-2016</SVFROMDATE>
    <SVTODATE TYPE=”Date”>1-Apr-2016</SVTODATE>
</STATICVARIABLES>
```

Gathering a customized report_

When a required report is not available at Tally end, the TDL code required for the custom report can be sent through the XML request to get the information.

Tag **<TDL>**

The tag <TDL> is used to specify the TDL related information. This particular Tag is specified only when the TDL Code required to serve the request and which is not present in default code of TallyPrime. The complete TDL to be executed in order to handle the Request; The TDL will be sent within the TDL block. Tally application will respond depending on the TDL request.

Example: A Request for a report 'Simple Trial Balance' has to be made. But, 'Simple Trial balance' is not present in default code, thus, the definition of the same is specified in the TDL tag.

The TDL program is sent using TDL tag as per the following structure:

```
<TDL>

<TDLMESSAGE>

TDL request specification

</TDLMESSAGE>

</TDL>
```

The <**TDLMESSAGE**> tag is mandatory inside the <TDL> tag. Inside that, we can write all the definitions and its attributes with their values. All TDL definitions and attributes are represented as tags.

Report, Collection, Function and Object can be specified inside the <TDL> tag. Consider the following examples, which demonstrate the usage of <HEADER> values:

Report Specification in TDL

```
<HEADER>

  <VERSION>1</VERSION>

  <TALLYREQUEST>Export</TALLYREQUEST>

  <TYPE>Data</TYPE>

  <ID>Report Name</ID>

</HEADER>
```

In the above header format, the value of TallyRequest is **Export** and the Type is 'Data'. Therefore, the value of ID must be a name of a Report. This report name should be specified inside the tag <REPORT> within the <TDL> tag.

Example:

```
<TDL>

<TDLMESSAGE>

<REPORT NAME="TDL Report" ISMODIFY="No" ISFIXED="No" ISINITIALIZE="No" ISOPTION="No"
ISINTERNAL="No">

<FORMS>First TDL Form</FORMS>

</REPORT>

                    .
```

```
              .
</TDLMESSAGE>
</TDL>
```

Let’s illustrate this case by sending a request to fetch data from the report ‘Simple Trial Balance’ that doesn’t exist at Tally end.

- The XML Request with the TDL code for the customized report ‘Simple Trial Balance’

```
<ENVELOPE><HEADER>
<VERSION>1</VERSION>
<TALLYREQUEST>Export</TALLYREQUEST>
<TYPE>Data</TYPE>
<ID>Simple Trial balance</ID>
</HEADER>
<BODY>
<DESC>
  <STATICVARIABLES>
   <EXPLODEFLAG>Yes</EXPLODEFLAG>
<SVEXPORTFORMAT>$$SysName:XML</SVEXPORTFORMAT>
  </STATICVARIABLES>
<TDL>
<TDLMESSAGE>
<REPORT NAME=”Simple Trial balance”>
 <FORMS>Simple Trial balance</FORMS>
 <TITLE>”Trial Balance”</TITLE>
</REPORT>
<FORM NAME=”Simple Trial balance”>
<TOPPARTS>Simple TB Part</TOPPARTS>
<HEIGHT>100% Page</HEIGHT>
<WIDTH>100% Page</WIDTH>
</FORM>
<PART NAME=”Simple TB Part”>
 <TOPLINES>
```

```
      Simple TB Title, Simple TB Details
   </TOPLINES>
   <REPEAT>Simple TB Details : Simple TB Ledgers </REPEAT>
   <SCROLLED>Vertical</SCROLLED>
   <COMMONBORDERS>Yes</COMMONBORDERS>
</PART>
<LINE NAME=”Simple TB Title”>
    <USE>Simple TB Details</USE>
    <LOCAL>Field : Default : Type : String    </LOCAL>
    <LOCAL>Field : Default : Align : Centre      </LOCAL>
    <LOCAL>Field : Account Name : Set as: “Particulars” </LOCAL>
    <LOCAL>Field : Account Value: Set as: “Amount”</LOCAL>
    <BORDER>Flush Totals</BORDER>
</LINE>
<LINE NAME=”Simple TB Details”>
    <LEFTFIELDS>Account Name</LEFTFIELDS>
    <RIGHTFIELDS>Account Value</RIGHTFIELDS>
    <XMLTAG>”Accts Info”</XMLTAG >
</LINE>
<FIELD NAME=”Account Name”>
    <USE>Account Name</USE>
    <SET>$Name</SET>
</FIELD>
<FIELD NAME=”Account Value”>
    <USE>Account Value</USE>
    <SET>$ClosingBalance</SET>
    <BORDER>Thin Left</BORDER>
</FIELD>
<COLLECTION NAME=”Simple TB Ledgers”>
     <TYPE>Ledger</TYPE>                                                   <FILTERS>NoProfitsimple</FILTERS>
</COLLECTION>
<SYSTEM TYPE=”Formulae” NAME=”NoProfitSimple”>
```

```
    NOT $$IsLedgerProfit
</SYSTEM>
</TDLMESSAGE>
</TDL>
</DESC>
</BODY>
</ENVELOPE>
```

The above XML Request is similar to the previous request given for getting the report Trial Balance. The difference here is the Report Name contained within the Tag “<ID>” is not specified in Tally. Therefore, In the Tag <BODY> within Tag <DESC>, an additional tag <TDL> must be specified with the TDL describing the Report and its components enclosed within Tag <TDLMESSAGE>.’

XML Response Received

```
<ENVELOPE>
 <ACCTSINFO>
  <ACCOUNTNAME>Abc Company Limited</ACCOUNTNAME>
  <ACCOUNTVALUE>1,29,377.00</ACCOUNTVALUE>
 </ACCTSINFO>
  <ACCOUNTNAME>Sales</ACCOUNTNAME>
  <ACCOUNTVALUE>1,29,277.00</ACCOUNTVALUE>
 </ACCTSINFO>
    …
    …
    …
</ENVELOPE>
```

> The feature ‘Convert to XML TDL’ in TallyPrime Developer allows you to convert projects/files that are in TDL to XML TDL. This simplifies the generation of XML request.

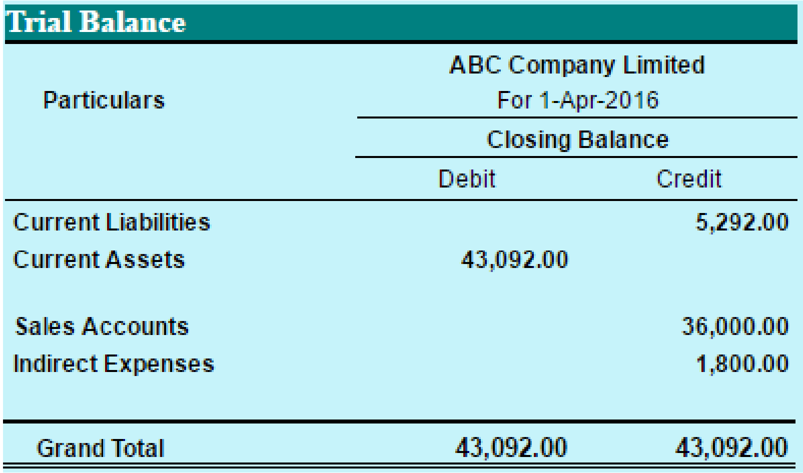
Gathering a report in HTML format

Apart from XML format, the reports of Tally can be gathered in HTML format as well. The below mentioned XML request gets the report ‘Trial Balance’ in HTML format:

```
<ENVELOPE>
<HEADER>
   <VERSION>1</VERSION>
   <TALLYREQUEST>Export</TALLYREQUEST>
```

```
  <TYPE>Data</TYPE>
  <ID>Trial Balance</ID>
</HEADER>
<BODY>
  <DESC>
         <STATICVARIABLES>
           <SVEXPORTFORMAT>$$SysName:HTML</SVEXPORTFORMAT>
       </STATICVARIABLES>
  </DESC>
</BODY>
</ENVELOPE>
```

The below mentioned output/screen is the response received in HTML format.



Trial Balance

Particulars	ABC Company Limited For 1-Apr-2016 Closing Balance	
	Debit	Credit
Current Liabilities		5,292.00
Current Assets	43,092.00	
Sales Accounts		36,000.00
Indirect Expenses		1,800.00
Grand Total	43,092.00	43,092.00

The HTML content of the response has to be saved in an html file (*.html) to view the content as shown above.

Guidelines

Names of some most frequently used reports in Tally

Accounting Reports

Name of the Report	Description of the report	Mandatory Variables	Purpose of mandatory variables
Day Book	Details of vouchers for a specified period		

Trial Balance	Opening / transacted / Closing Balances of all accounting groups and ledgers		
Ledger Vouchers	Ledger Account for a ledger	LedgerName	To set the Name of the ledger
Ledger Outstandings	Outstanding report for a ledger	LedgerName	To set the Name of the ledger
Bills Payable	Outstanding report for Payables		
Bills Receivable	Outstanding report for Receivables		
Group Outstandings	Outstanding report for a group	GroupName	To set the Name of the group

Inventory Reports

Common Variables used in Tally Reports

Name of the Variable	Data Type	Permissible value	Description
SVFROMDATE	Date	Uni date in format of YYYYMMDD	To provide from-date of the period
SVTODATE	Date	Uni date in format of YYYYMMDD	To provide the To-date of the period
SVEXPORTFORMAT	String	$$SysName:XML $$SysName:HTML	To set the format of the report
EXPLODEFLAG	Logical	Boolean	To get the report in the Detailed or

			Condensed Mode

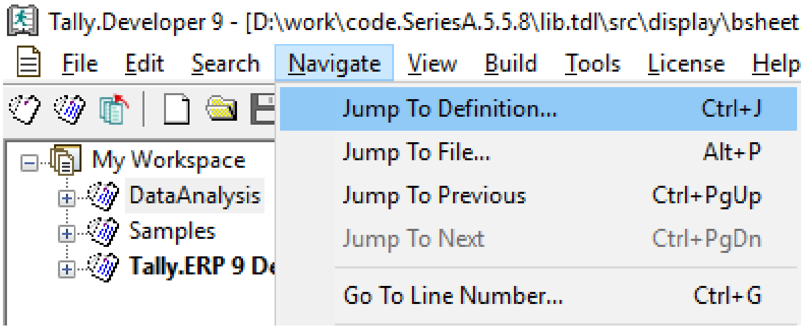
Note: All the other variables used across the reports in Tally can be referred in default TDL which is available in TallyPrime Developer.

Finding the report name/code in default TDL

The TDL code for the required Tally report can be viewed in TallyPrime Developer.

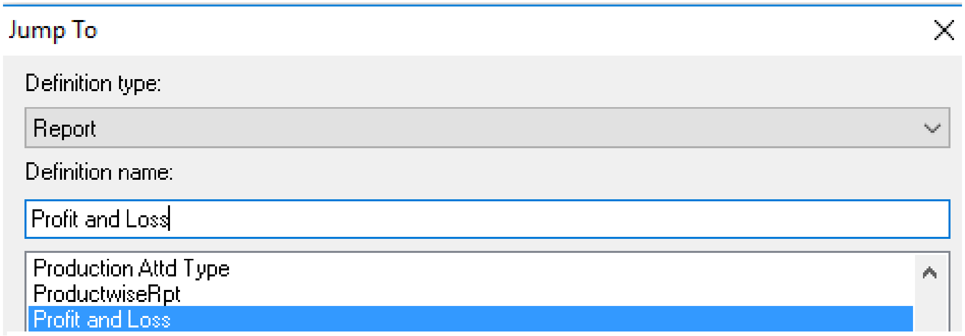
Example: To find the TDL code of the Tally report ‘Profit and Loss’

Step-1: Click on menu ‘Navigate’ and select the option ‘Jump to Definition’


Tally.Developer 9 - [D:\work\code.SeriesA.5.5.8\lib.tdl\src\display\bsheet
File Edit Search Navigate View Build Tools License Help
Jump To Definition... Ctrl+J
Jump To File... Alt+P
Jump To Previous Ctrl+PgUp
Jump To Next Ctrl+PgDn
Go To Line Number... Ctrl+G
My Workspace
DataAnalysis
Samples
Tally.ERP 9 De

Step-2: Select **Report** from this definition list shown in field **Definition Type**

Step-3: Type **Profit and Loss** and select the report from the list of reports shown in field **Definition Name**


Jump To

Definition type:

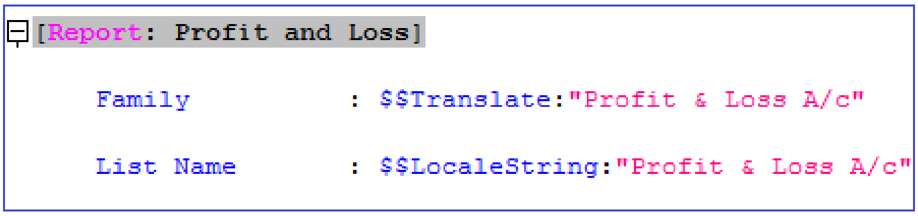
Report

Definition name:

Profit and Loss

Production Attd Type
ProductwiseRpt
Profit and Loss

The cursor will be jumped to the selected definition as per the example as given below:


```
[Report: Profit and Loss]

     Family          : $$Translate:"Profit & Loss A/c"

     List Name       : $$LocaleString:"Profit & Loss A/c"
```

Related Topics

- How to Generate e-Invoice Online Using TallyPrime (KSA e-Invoice Integration Phase)
- e-Invoicing Under KSA e-Invoice Integration Phase - FAQ
- What's new in TDL with regards to WhatsApp integration?
- Developer Guide for KSA e-Invoicing Integration Phase
- Integration Methods and Technologies

Have another query?
Chat with us on +91 9019910043