

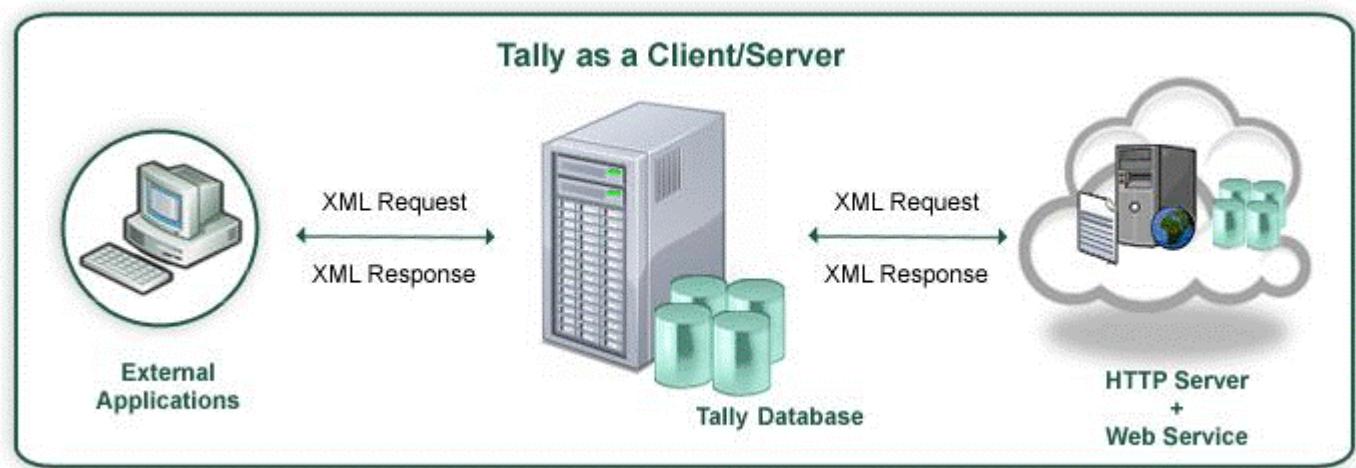
Integration using XML

Last updated on: August 25, 2025 | 6 min read

TallyPrime enables integration with various web scripting languages such as ASP, Perl, PHP, VB, and any platform that can support XML over HTTP. This capability is built-in, making it easy to import and export data in XML format.

TallyPrime delivers web service-like functionality by exposing its internal data and operations via XML over HTTP. It can communicate with any system capable of sending or receiving XML messages, allowing it to function as both a server and a client.

This section provides a foundational understanding of how XML messaging works in TallyPrime and outlines the standard message structure used for integration.



XML Schema Overview and Base Document Structure

TallyPrime uses a consistent XML structure for communication, built around an **ENVELOPE** element that contains a **HEADER** and **BODY**.

Components of Request or Response

<ENVELOPE> is the top element of the XML fragment which is representing the message. Both Request and Response consists of two sections:

- Header
- Body

Header Information

Header section will give all identification information to the recipient such as authentication, transaction management, and payment so on. This section determines how the recipient of the

message should process the information. Header information is classified in two ways, one is for Request and the other is for Response. All the information about Request or Response is enclosed with Header Tags.

In case of Request, header information includes mainly four elements which are Version, TallyRequest, Type and ID. Version gives the version of the message format. Second element TallyRequest will identify the type of request as Import or Export in the messaging format. If the value of Tally Request is Import then the type of information would be Data, and the request will be identified by the report name specified in ID. If the value of Tally Request is Export then the type of information would be Data, Collection, Object or Function. The ID specifies the name of Report, Collection, Object or function.

In the case of Response, there are mainly two elements which are Version and Status. Version gives the version of the message format. Status indicates whether the request is success or failure.

Body Information

It exchanges the information intended for the recipient of the message. This section gives the actual details of the message. It is further divided into two sections:

- Description for Request/Response
- Data required for the Request/Response

Description section is used to give the description for message, request or response. Description element mainly includes all types of variable information, storage information, computational information and user defined TDLs. All the description information is enclosed with <DESC> tags.

Data section includes all the data information being transferred. All the data should be enclosed within the <DATA> tags.

Basic Template:

```
<ENVELOPE>
  <HEADER>
    <TALLYREQUEST>Import Data</TALLYREQUEST>
  </HEADER>
  <BODY>
    <IMPORTDATA>
      <REQUESTDESC>
        <REPORTNAME>Vouchers</REPORTNAME>
      </REQUESTDESC>
      <REQUESTDATA>
        <TALLYMESSAGE>
          <!- XML data for vouchers, ledgers, etc. ->
        </TALLYMESSAGE>
      </REQUESTDATA>
    </IMPORTDATA>
  </BODY>
</ENVELOPE>
```

```
</TALLYMESSAGE>  
</REQUESTDATA>  
</IMPORTDATA>  
</BODY>  
</ENVELOPE>
```

Explanation of Common XML Tags

Header Tags

- **<VERSION>**: Indicates the version of the messaging format.
- **<TALLYREQUEST>**: Defines the nature of the request. Supported values: Import, Export, or Execute.
- **<TYPE>**: Specifies the category of data or action. Can be **DATA**, **COLLECTION**, **OBJECT**, **FUNCTION**, or **ACTION**.
- **<SUBTYPE>**: Optional field to further define the type.
- **<ID>**: Identifies the specific object or collection.
- **<STATUS>**: Present in responses. A value of 1 means success; 0 indicates failure.

Body Tags

- **<DESC>**: Describes additional parameters for the request.
 - **<STATICVARIABLES>**: Includes global variables like **SVCURRENTCOMPANY**, **SVFROMDATE**, etc.
 - **<REPEATVARIABLES>**: Specifies ranges or repeated inputs.
 - **<FETCHLIST>**: Lists the fields or methods to be retrieved.
 - **<FUNCPARAMLIST>**: Lists function parameters.
 - **<TDL>**: Contains embedded TDL code if the request involves dynamic logic.
- **<DATA>**: Carries the actual data being transferred or received.

Tally XML Gateway Configuration for Communication

To enable XML-based integration in TallyPrime:

1. Open TallyPrime and go to **F1 (Help) → Settings → Advanced Configuration**.
2. Enable the **HTTP Server**. The default port is 9000.
3. If needed, also enable the **ODBC Server**.
4. External applications can post XML to this endpoint: <http://<Tally-IP>:9000>

File-based XML imports are also supported by placing XML files into the appropriate Tally import directory.

SOAP Envelope Examples for Service Calls

Although TallyPrime does not use SOAP natively, it supports XML payloads that can be wrapped inside a SOAP envelope when used with middleware tools.

SOAP-Wrapped Example:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ENVELOPE>
      <HEADER>
        <TALLYREQUEST>Import Data</TALLYREQUEST>
      </HEADER>
      <BODY>
        <IMPORTDATA>
          <REQUESTDESC>
            <REPORTNAME>Vouchers</REPORTNAME>
          </REQUESTDESC>
          <REQUESTDATA>
            <TALLYMESSAGE>
              <VOUCHER VCHTYPE="Sales" ACTION="Create">
                <DATE>20240617</DATE>
                <PARTYLEDGERNAME>Customer A</PARTYLEDGERNAME>
                <AMOUNT>10000</AMOUNT>
              </VOUCHER>
            </TALLYMESSAGE>
          </REQUESTDATA>
        </IMPORTDATA>
      </BODY>
    </ENVELOPE>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML Message Formats and Templates

TallyPrime follows a standardized XML message format comprising a header and body enclosed in an envelope.

Request Template

```
<ENVELOPE>
  <HEADER>
    <VERSION>1</VERSION>
    <TALLYREQUEST>Export</TALLYREQUEST>
    <TYPE>DATA</TYPE>
    <ID>All Masters</ID>
  </HEADER>
  <BODY>
    <DESC>
      <STATICVARIABLES>
        <SVCURRENTCOMPANY>ABC Company Ltd</SVCURRENTCOMPANY>
      </STATICVARIABLES>
      <FETCHLIST>
        <FETCH>Name</FETCH>
        <FETCH>ClosingBalance</FETCH>
      </FETCHLIST>
    </DESC>
  </BODY>
</ENVELOPE>
```

Response Template

```
<ENVELOPE>
  <HEADER>
    <VERSION>1</VERSION>
    <STATUS>1</STATUS>
  </HEADER>
  <BODY>
    <DATA>
      <!-- Returned records -->
    </DATA>
```

```
</BODY>
```

```
</ENVELOPE>
```

Failure Template

```
<ENVELOPE>
```

```
<HEADER>
```

```
<VERSION>1</VERSION>
```

```
<STATUS>0</STATUS>
```

```
</HEADER>
```

```
<BODY>
```

```
<DATA>
```

```
<STATUS.LIST>
```

```
<STATUS>
```

```
<CODE>123</CODE>
```

```
<DESC>Invalid Request</DESC>
```

```
</STATUS>
```

```
</STATUS.LIST>
```

```
</DATA>
```

```
</BODY>
```

```
</ENVELOPE>
```

This consistent message structure ensures reliable data communication between TallyPrime and external systems.

