

Integration Methods and Technologies

Last updated on: November 17, 2025 | ⏳ 9 min read

TallyPrime supports a variety of integration methods to enable seamless connectivity with external systems. These methods range from built-in options using TDL (Tally Definition Language) to advanced integrations using external applications or libraries.

Integration Using TDL

Tally Definition Language (TDL) is Tally's native language used to customize the UI, reports, and business logic within TallyPrime. It is tightly integrated with the product and is ideal for in-product automation and data handling.

Why TDL?

- **Purpose-built for TallyPrime**, ensuring seamless customization.
- Enables **custom UI, reports, and validations** within the application.
- Supports **automation of internal workflows** like auto-filling data, triggering alerts, or validating entries.
- Offers **real-time deployment** of changes without the need for external tools.

Where TDL Fits

TDL is best suited for **in-product customizations** and **defining data structures** for import/export. While external systems may handle API calls or file transfers, TDL manages the **data flow within TallyPrime**, acting as the bridge between the UI, data, and external integrations.

Key Capabilities of TDL Integration

- **Seamless Customization within TallyPrime:** Create reports, modify screens, or extend features directly inside Tally's UI without relying on external apps.
- **Direct Access to Tally Data:** Gain structured access to masters, vouchers, ledgers, and more—enabling precise control over display, validation, and manipulation.
- **Real-Time Data Sync:** TDL can call external APIs or use ODBC/XML to push/pull data—enabling near real-time integrations.
- **High Performance:** Being native and compiled within Tally, TDL ensures optimal performance without network latency or third-party overhead.
- **Secure and Role-Based Execution:** Operates within Tally's environment, ensuring built-in access control and secure data operations.
- **Cost-Effective and Scalable:** Ideal for incremental enhancements—no additional server, middleware, or platform dependency.

- **Embedded Business Logic:** TDL allows validation, conditional flows, and automation (e.g., auto voucher creation) directly in sync with business rules.

JSON Integration Using TDL

TDL also supports integration with external systems that use JSON for data exchange. Developers can use TDL to both fetch and export structured JSON data using HTTP or local file sources.

Key Capabilities

HTTP Integration

- **External API to TallyPrime Using Native JSON Format:** External API initiates HTTP POST where JSON data is in the native Tally structure that can be directly posted to TallyPrime. And HTTP Get to fetch data in native format or custom/non-native format using report attributes.
 - JSON Request Response Formats & Variable Setting
 - [JSON Tag as Desc Name](#)
 - [JSON Tag](#)
 - Report Attributes
 - [Export Empty Fields](#)
 - [Export Header](#)
 - [Plain JSON](#)
 - Functions
 - [\\$\\$HTTPInfo](#)
 - [\\$\\$ImportInfo](#)
 - [\\$\\$LastImportError](#)
 - [\\$\\$ImportType](#)
 - [\\$\\$ImportAction](#)
 - Import File & Attributes
 - [Response Report](#)
 - Import File Events
 - [After Import Object](#)
- **TallyPrime to External API Using Custom JSON Format:** TallyPrime uses TDL to fetch JSON from external APIs via HTTP GET and POST, where the report capabilities are used to construct the json requests along with headers.
 - Collection Attributes

- [DataSource: HTTPJSONEX](#)
- [DataSource:HTTPJSON](#)
- [Export Header](#)
- [Remote Request](#)
- [JSON Object Path](#)
- [JSON Object](#)
- General Attributes
 - [JSON Tag as Desc Name](#)
 - [JSON Tag](#)
- Report Attributes
 - [Export Empty Fields](#)
 - [Export Header](#)
 - [Plain JSON](#)
- Actions
 - [HTTP Request](#)
- Functions
 - [\\$\\$GetMethodNameValue](#)

File Based Integration

- **Reading JSON File:** Read JSON files stored locally into TallyPrime by mapping JSON objects to TDL collections. Following attributes are used to do the same.
 - Collection Attributes
 - [DataSource: FileJSON](#)
 - [DataSource: JSONString](#)
 - [JSON Object Path](#)
 - [JSON Object](#)
 - Functions
 - [\\$\\$GetMethodNameValue](#)
- **File Based Export:** Default and Customized Reports are exported using export feature of TallyPrime or Export action. Following attributes are used to create the required JSON structure to be exported.
 - General Attributes
 - [JSON Tag as Desc Name](#)
 - [JSON Tag](#)

- Report Attributes
 - Export Empty Fields
 - Plain JSON
- Action
 - Export
- **JSON Data export using Productivity Suites:** Data can be exported to a JSON schema present in a file by providing mapping information in the Form. Following attributes are used for the same.
 - Form Attributes
 - Resource
 - Map/JSON Map
 - Resource Attribute
 - Source
 - Resource Type: JSON
 - Functions
 - \$\$TplLine
 - \$\$WordInfo
 - \$\$ExcelInfo
 - \$\$IsFileTypeSupported
- **File Based Import in Native JSON Format:** JSON file where the data is in native structure which can be directly imported using import feature of TallyPrime or with the Import action. Following are the attributes that can be used to log information or modify objects during import.
 - Functions
 - \$\$ImportInfo
 - \$\$LastImportError
 - \$\$ImportType
 - \$\$ImportAction
 - Action
 - Import
 - Import File & Attributes
 - Empty
 - Import File Events

- Start Import
- Import Object
- After Import Object
- End Import

These capabilities allow Tally to integrate with a wide range of modern applications and services that rely on JSON for communication.

XML Integration Using TDL

Tally Definition Language (TDL) allows TallyPrime to communicate with external systems through XML. This integration enables Tally to either **fetch data from** or **send data to** remote web services and databases.

Key Capabilities

HTTP Integration

- **External API to TallyPrime using Native XML Format:** External API initiates HTTP POST where XML data is in the native Tally structure that can be directly posted to TallyPrime. And HTTP Get to fetch data in native format or custom/non-native format using report attributes.
 - XML Request Response Formats & Variable Setting
 - General Attributes
 - XML Tag
 - Report Attributes
 - Export Header
 - Plain XML
 - Functions
 - \$\$HTTPInfo
 - \$\$ImportInfo
 - \$\$LastImportError
 - \$\$ImportType
 - \$\$ImportAction
 - Import File & Attributes
 - Response Report
 - Import File Events
 - After Import Object
- **TallyPrime to External API using Custom XML format:** TallyPrime uses TDL to fetch XML from external APIs via HTTP GET and POST, where the report capabilities are used to construct the

XML requests along with headers.

- Collection Attributes
 - DataSource:HTTPXML
 - Remote URL
 - Export Header
 - Remote Request
 - XML Object Path
 - XML Object
- General Attributes
 - XML Tag
- Report Attributes
 - Export Header
 - Plain XML
- Actions
 - HTTP POST
 - HTTP Request
- Functions
 - \$\$GetMethodNameValue

File Based Integration

- **Reading XML File:** Read XML files stored locally into Tally by mapping XML objects to TDL collections. Following attributes are used to do the same.
 - Collection Attributes
 - DataSource: FileXML
 - DataSource: XMLString
 - XML Object Path
 - XML Object
 - Functions
 - \$\$GetMethodNameValue
- **File Based Export:** Default and Customized Reports are exported using export feature of TallyPrime or Export action. Following attributes are used to create the required XML structure to be exported.
 - General Attributes

- XML Tag
- Report Attributes
 - Export Empty Fields
 - Plain XML
- Action
 - Export
- **XML Data export using Productivity Suites:** Data can be exported to an XML schema present in a file by providing mapping information in the Form. Following attributes are used for the same.
 - Form Attributes
 - Resource
 - Map/XML Map
 - Resource Attribute
 - Source
 - Resource Type: XML
 - Functions
 - \$\$TplLine
 - \$\$WordInfo
 - \$\$ExcelInfo
 - \$\$IsFileTypeSupported
- **File Based Import in Native XML Format:** XML file where the data is in native structure which can be directly imported using import feature of TallyPrime or with the Import action. Following are the attributes that can be used to log information or modify objects during import.
 - Functions
 - \$\$ImportInfo
 - \$\$LastImportError
 - \$\$ImportType
 - \$\$ImportAction
 - Action
 - Import
 - Import File & Attributes
 - Empty

- Import File Events
 - Start Import
 - Import Object
 - After Import Object
 - End Import

Key Use Cases of XML & JSON Integration

1. Show External Data in TallyPrime

- TallyPrime sends a GET Request to get data from external application and display as report
- TallyPrime reads a JSON file and displays data as report
- TallyPrime sends a POST request to external application and shows the status response in a message box or a log.

2. User-Driven Data Retrieval

- Tally accepts input from the user (like an item name), uses it to create a custom XML or JSON request, and sends it to fetch related data from an external system.

3. Control Master/Voucher Creation

- While Importing masters or vouchers from XML or JSON data, TDL allows control over whether new masters or vouchers should be created and also allows to modify some data before importing

DLL Integration with TallyPrime

DLLs allow Tally to interact with external databases, cloud services, and hardware devices that are beyond the scope of TDL.

Key Capabilities

- **Integration in JSON Format:** TallyPrime interacts with the external DLLs where the data exchange happens in JSON Format
 - Collection Attributes
 - DataSource: Plugin JSON
 - JSON Object Path
 - JSON Object
 - Input JSON
 - Input Parameter
 - Break On

- Functions
 - \$\$GetMethodNameValue
- General Attributes
 - JSON Tag as Desc Name
 - JSON Tag
- Report Attributes
 - Export Empty Fields
 - Plain JSON
- **Integration in XML Format:** TallyPrime interacts with the external DLLs where the data exchange happens in XML Format
 - Collection Attributes
 - DataSource: Plugin XML
 - DataSource: Ax Plugin XML
 - XML Object Path
 - XML Object
 - Input XML
 - Input Parameter
 - Break On
 - Functions
 - \$\$GetMethodNameValue
 - General Attributes
 - XML Tag
 - Report Attributes
 - Export Empty Fields
 - Plain XML

Key Use Cases

- Connect to SQL Server, MySQL, or other relational databases.
- Interact with external systems such as ERP platforms or cloud APIs.
- Integrate with hardware like barcode scanners or POS terminals.

DLLs can be invoked from TDL using custom functions and allow more control over external workflows, including database operations and device handling.

TDL vs External Integration

Customization Scope

With **TDL**, customization is deeply embedded within TallyPrime. Developers can directly modify the **UI, reports, vouchers, masters, and workflows**. This means you can not only fetch or send data but also design how that data appears in Tally, apply validations, and even create entirely new reports. For example, you can add a custom button in a voucher screen that fetches real-time stock prices via JSON API and displays them immediately inside Tally.

On the other hand, **external integrations** (XML schema, SQL connections, DLLs) cannot change Tally's interface or reports. Their role is limited to **data exchange**—either importing structured XML into Tally, exporting vouchers/ledgers, or syncing with external systems. The customization happens outside of Tally, in the external app or middleware, not inside Tally itself.

Business Logic

In **TDL**, business logic is executed **within Tally's environment**. This allows validations, conditional data flows, and automation to happen at the point of entry. For instance, a TDL customization can prevent a user from saving a sales voucher if credit limits are exceeded, or automatically create an accounting entry when a stock movement occurs. The logic is tightly coupled with Tally's data model and user actions.

With **external integrations**, business logic usually resides in the external application, database, or API. Tally mostly acts as a consumer or provider of structured data (XML, JSON). For example, a SQL server might apply business rules to imported vouchers before they are inserted into Tally, or an ERP system may dictate how ledger balances are synced. In this setup, Tally doesn't enforce the rules—the external system does.



Use TDL for workflows and reports inside Tally. Use external integration methods for connecting Tally with other platforms, services, or devices.

Have another query?

Scan and Chat

OR "Save the number +91 9019910043 to chat with us"



» On this
page