



POWER OF SIMPLICITY

Developer Manual – Auto Fill from PRN

The information contained in this document is current as of the date of publication and subject to change. Because Tally must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Tally, and Tally cannot guarantee the accuracy of any information presented after the date of publication. The information provided herein is general, not according to individual circumstances, and is not intended to substitute for informed professional advice.

This document is for informational purposes only. TALLY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT AND SHALL NOT BE LIABLE FOR LOSS OR DAMAGE OF WHATEVER NATURE, ARISING OUT OF, OR IN CONNECTION WITH THE USE OF OR INABILITY TO USE THE CONTENT OF THIS PUBLICATION, AND/OR ANY CONDUCT UNDERTAKEN BY PLACING RELIANCE ON THE CONTENTS OF THIS PUBLICATION.

Complying with all applicable copyright and other intellectual property laws is the responsibility of the user. All rights including copyrights, rights of translation, etc., are vested exclusively with TALLY SOLUTIONS PRIVATE LIMITED. No part of this document may be reproduced, translated, revised, stored in, or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Tally Solutions Pvt. Ltd.

Tally may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written licence agreement from Tally, the furnishing of this document does not give you any licence to these patents, trademarks, copyrights, or other intellectual property.

© 2009 Tally Solutions Pvt. Ltd. All rights reserved.

Tally.Developer 9 is either registered trademarks or trademarks of Tally Solutions Pvt. Ltd. in India and/or other countries. All other trademarks are properties of their respective owners.

Version: Developer Manual – Auto Fill from PRN/1.0/October 2009

Auto Fill from PRN

Scenario

A firm **Global Enterprises** using Tally.ERP 9 is making use of order processing. They have a voucher by name **Purchase Requisition** created under **Purchase Order**. They need to track **Purchase Requisition** against **Purchase Order** vouchers.

Requirement Statement

User will be making Purchase Requisition voucher. After making this voucher, user will track this voucher with Purchase Order. All the values from Purchase Requisition should get filled in Purchase Order.

Solution

In order to update values from Purchase Requisition to Purchase Order, the following steps have to be adhered:

I. Changes in Invoice

- a. Table of Purchase Requisition should be populated in Purchase Order. To display this Table we need to add a field to Purchase Order voucher. We would add an optional part, so that the field appears only if voucher is a Purchase Order

```
[#PART: EI Orders]

ADD : OPTIONS : EI PRNRefOrder No      : @@IsOnlyPuchOrder
```

- b. The collection should contain only those numbers used in Purchase Requisition for that party. Hence we need to create a collection and filter accordingly.

```
[COLLECTION : PRN Order]

TYPE          : Ledger Purchase Orders
COLLECTION    : Not Applicable
FORMAT       : $Name, 7
FORMAT       : $Date, 10
TITLE        : $$LocaleString:"PRN Order"
FILTER       : OnlyRequisition
CHILD OF     : $PartyLedgerName
```

- c. On selection of this table the values should get filled in the voucher. We have made use of **On: Form Accept** event at **EI Supplementary** Form and called a function.

```
[#FORM: EI Supplementary]

ON : Form Accept : @@IsOnlyPuchOrder : CALL : My VoucherFill : #PRNOrderDetails : #PRNOrderDate
```

- d. This function walks to a collection which will help us to get values; by making use of action **Set Value** we are setting values for the methods. We are also making use of action **Insert Collection Object** for moving down in the collection and **Set Target** to move up in the hierarchy.

```
[FUNCTION: My Voucher Fill]

PARAMETER : PRNNo : STRING
PARAMETER : PRNDate : DATE

00 : WALK COLLECTION : IE PRNVoucher
01 : INSERT COLLECTION OBJECT : Inventory Entries
02 : SET VALUE : Stock Item Name
03 : SET VALUE : Rate
04 : SET VALUE : Amount
05 : SET VALUE : IsDeemedPositive : No
06 : INSERT COLLECTION OBJECT : BATCHALLOCATIONS
07 : SET VALUE : OrderDueDate : $$DateRange:"10 Days":($TgtObject:$Date):True
08 : SET VALUE : ActualQty
09 : SET VALUE : BilledQty
10 : SET VALUE : Rate
11 : SET VALUE : Amount
12 : SET TARGET : ..
13 : END WALK
```

- e. The collection from which we need to fetch values to Purchase Order has to be created and filter value of the collection, based on the parameters passed for the function.

```
[COLLECTION: My PRN Vouchers]

TYPE : Vouchers : VoucherType
CHILD OF: "Purchase Requisition"
FILTER : My Voucher Filter
```

- f. Values we require to fetch in the voucher also has inventory entries and batch allocations, hence we would not use the above collection. However we would use it as a source collection in a summary collection and walk to Batch Allocations

```
[COLLECTION: IE PRNVoucher]

SOURCE COLLECTION : My PRN Vouchers
WALK : Inventory Entries, Batch Allocations
BY : StockItemName : $StockItemName
AGGR COMPUTE : ActualQty : SUM : $ActualQty
AGGR COMPUTE : BilledQty : SUM : $BilledQty
AGGR COMPUTE : Amount : SUM : $Amount
COMPUTE : Rate : ($$AsRate:($Amount/$BilledQty))
```

TDL Capabilities Used

- I. Modified Part to add optional Part
- II. 'On : Form Accept' event used at Form Level
- III. Creation of a User Defined Function. Actions used in User Defined Functions
 - a. Walk Collection
 - b. Insert Collection Object
 - c. Set Value
 - d. Set Target
- IV. Attributes used at collection level
 - a. Child Of
 - b. Filter
 - c. Source Collection
 - d. Walk
 - e. By
 - f. Aggr Compute
 - g. Compute