enactor®
retail systems for a digital world

How-To Guide - Transaction Recovery

Document Type: External

Document Version: 1.0

Date: 24/01/2024

Author: John Wood

Table of Contents

Introduction

The Transaction Recovery functionality allows a copy of an in-progress transaction to be stored in the local database each time the transaction is updated (e.g. when an item is added to it). In the event of a power cut or a POS restart, this function allows the current transaction to be recovered and either completed, stored, or voided.

There are numerous configuration options offered to control the behaviour of this function detailed in the User Roles section below. For example, it is possible to restrict the user to only be able to resume transactions that they originally created. Alternatively, it is possible to allow the user to resume transactions that were created by any user.

Enactor will store both Retail and Cash Management transactions. Transactions are written locally to the POS database in the *storedrecoverytransactions* table.

Recovered Cash Management transactions are automatically voided when the user signs in

Overview

This guide will cover the configuration for the following: (List down the key elements of the document)

- **POS Terminals** – Enables the function at the POS (or mobile POS)
- **User Roles** – Controls the behaviour of the Transaction Recovery function depending on the privileges assigned to users.

Prior Training / Experience

You should be familiar with the following:

- Estate Manager Configuration
- Data Broadcasting

Configuration Steps

There are two key configuration elements to enable and control the behaviour of Transaction Recovery.

Firstly, it must be enabled via the POS Terminal (or POS Terminal Template). Once enabled, privileges can be assigned to User Roles to control how the function behaves. The User Role configuration can also be used to define different behaviours for different User types – for example, a sales assistant may not have permission to void a recovered transaction, but a manager does.
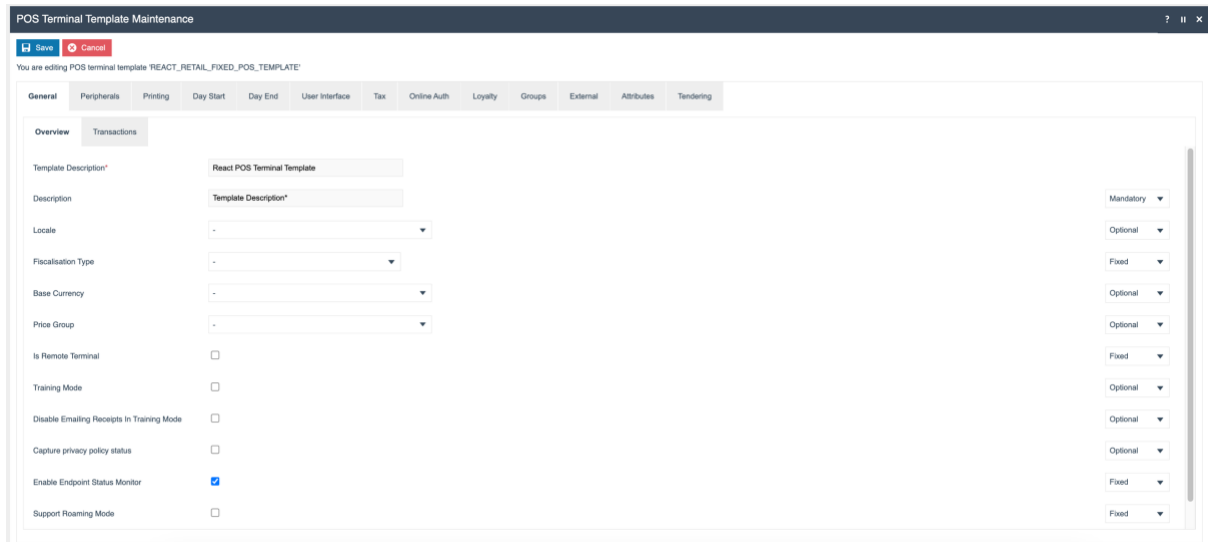
POS Terminal Template Configuration

This section describes configuring Automatic Transaction Recovery in the POS Terminal Template. The configuration steps detailed are equally applicable to a POS Terminal and could alternatively be defined there.

The POS Terminal Template maintenance application can be accessed through: Configuration-> Organisation -> POS Terminal Templates
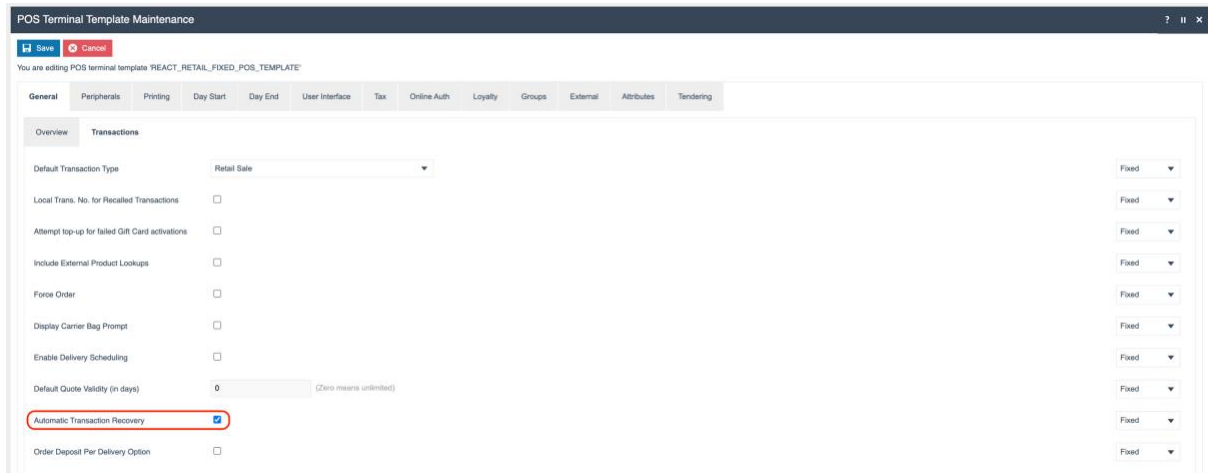

To edit the configuration of a POS Terminal Template, select the Edit icon (pencil) next to it.

Select the General > Transactions sub-tab

Tick the **Automatic Transaction Recovery** checkbox (and optionally set it to Fixed so this value applies to all POS Terminals that use this template), then select the Save icon to save the record. This completes the POS Terminal Template Configuration.

User Roles

The User Roles maintenance application allows a set of privileges to be assigned to a User Role, of which one or more can be assigned to a User.
This configuration makes the assumption that the Enactor POS Jar has been deployed via the Jar Deployer application so the Enactor POS privileges are available.

The User Role maintenance application can be accessed through:
Configuration -> HR -> User Roles

Authorisation to run the Transaction Recovery functionality is controlled with privileges maintained in User Roles.

Select the appropriate Role to edit, in this example, the SALES_ASSISTANT Role is being edited.

The privileges available to control the behaviour of transaction recovery are detailed in the following table. The Application Package for all privileges is Enactor POS.

Process	Function ID	Description
Handle Recovery Transactions	enactor.pos.AuthorisesSkipRecovery Transactions	Authorises the user to skip the transaction recovery screen at sign-on.
Handle Recovery Transactions	enactor.pos.AutoRecoverSingleTrans action	When a single recovery transaction is stored, it is automatically recovered at sign-on.
Handle Recovery Transactions	enactor.pos.ResumeRecoveryTransa ctionAllowed	Allows the user to resume a recovery transaction that they created.
Handle Recovery Transactions	enactor.pos.ResumeRecoveryTransa ctionForDifferentUserAllowed	Allows the user to resume a recovery transaction that any user created.
Handle Recovery Transactions	enactor.pos.SkipRecoveryTransactio nsAllowed	Allows the user to skip any recovery transactions.
Handle Recovery Transactions	enactor.pos.StoreRecoveryTransacti onAllowed	Allows the user to store a recovery transaction that they created.
Handle Recovery Transactions	enactor.pos.StoreRecoveryTransacti onForDifferentUserAllowed	Allows the user to store a recovery transaction that any user created.
Handle Recovery Transactions	enactor.pos.VoidRecoveryTransactio nAllowed	Allows the user to void a recovery transaction that they created.
Handle Recovery Transactions	enactor.pos.VoidRecoveryTransactio nForDifferentUserAllowed	Allows the user to void a recovery transaction that any user created.
Recover And Resume Transaction	enactor.pos.AuthorisesResumeReco veryTransaction	Authorises the user to resume a recovery transaction that they created.
Recover And Resume Transaction	enactor.pos.AuthorisesResumeReco veryTransactionForDifferentUser	Authorises the user to resume a recovery transaction that any user created.
Recover And Store Transaction	enactor.pos.AuthorisesStoreRecover yTransaction	Authorises the user to store a recovery transaction that they created.
Recover And Store Transaction	enactor.pos.AuthorisesStoreRecover yTransactionForDifferentUser	Authorises the user to store a recovery transaction that any user created.
Recover And Void Transaction	enactor.pos.AuthorisesVoidRecovery Transaction	Authorises the user to void a recovery transaction that they created.
Recover And Void Transaction	enactor.pos.AuthorisesVoidRecovery TransactionForDifferentUser	Authorises the user to void a recovery transaction that any user created.

Broadcasting

To deliver the configuration changes to the POS, broadcast the following entities:

- POS Terminals
- POS Terminal Templates
- Roles

POS Functionality

Transaction Recovery Functionality

The POS operator begins a transaction and adds some items. As each item is added, the transaction is updated in the *storedrecoverytransactions* table in the local database.

If the Enactor POS application is closed (this could be due to a reboot of the POS device, or a power cut), the next time it is launched, the operator is prompted to recover the in-progress transaction after they sign on:

In the scenario shown above, the POS operator doesn't have the privilege to auto-recover a single transaction, but does have the privileges to Resume, Store, Void and Skip the recovery transaction. Selecting the Resume function allows the transaction to be continued from where it left off.

When the transaction is completed, it is automatically purged from the *storedrecoverytransactions* table in the local database.

Host: 127.0.0.1 | Database: enactorpos | Query*

```
1    SELECT * FROM storedrecoverytransactions
```

Filter ...
- Columns
- SQL functions
- SQL keywords
- Snippets
- Query history
- Query profile

storedrecoverytransactions (0r × 13c)

ID	Entity_Name	Storage_Tag	Storage_Ref	User_Id	Storage_Date	Device_Id	Transaction_Type	Transaction_Total	Transaction_Currency_Id	Last_Updated	Training_Mode	XML

About This Document

©2024 Enactor Ltd

All documents produced by Enactor Ltd are supplied to customers subject to Copyright, commercial confidentiality and contractual agreements and must not be copied or transferred without permission.

The amendment history of this document can be found in the table below.

Current Document Version information

Document Version 1.0
Product Version 2.7

Document Context

This document is part of the Enactor Product Documentation Series. All Enactor products include a comprehensive documentation set designed to improve understanding of the product and facilitate ease of use.

Document Inquiries

At Enactor we aspire to producing the highest quality documentation to reflect and enhance the quality of our product. If you find that the document is inaccurate or deficient in any way, please assist us in improving our standard by letting us know.
For matters of document quality or any other inquiries regarding this document please contact:
By Email: documentation@enactor.co.uk

Document History

The following versions of the document have been produced:

VERSION	STATUS	ISSUE DATE	AUTHOR	REASON FOR ISSUE
1.0	Initial Draft	24/01/24	John Wood	