Error Handling Framework

* A standard exception handling framework supports persistence, retry and notification needs in an easy way for any exception handling requirement in the middleware layer.

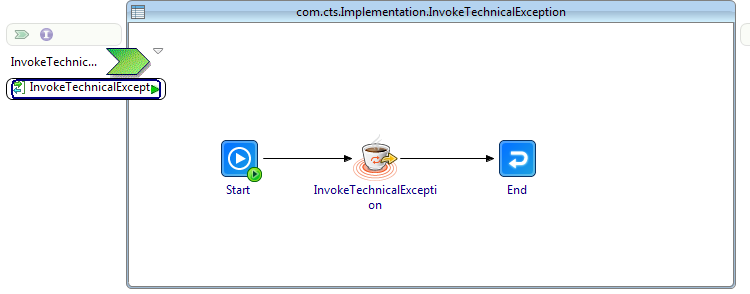
Artifacts:

* The solution is based on TIBCO BusinessWorks 6.2 and it has the following below features:
* Systematic logging.
* Persistence.
* Retry Mechanism.
* Single and Bulk Notification.
* The solution has two components – Exception Handling and Notification service.
* Dependencies:-
* TIBCO BusinessWorks 6.2
* TIBCO EMS Server 6.1 or higher
* TEA Admin 2.1
* Oracle 10g or higher

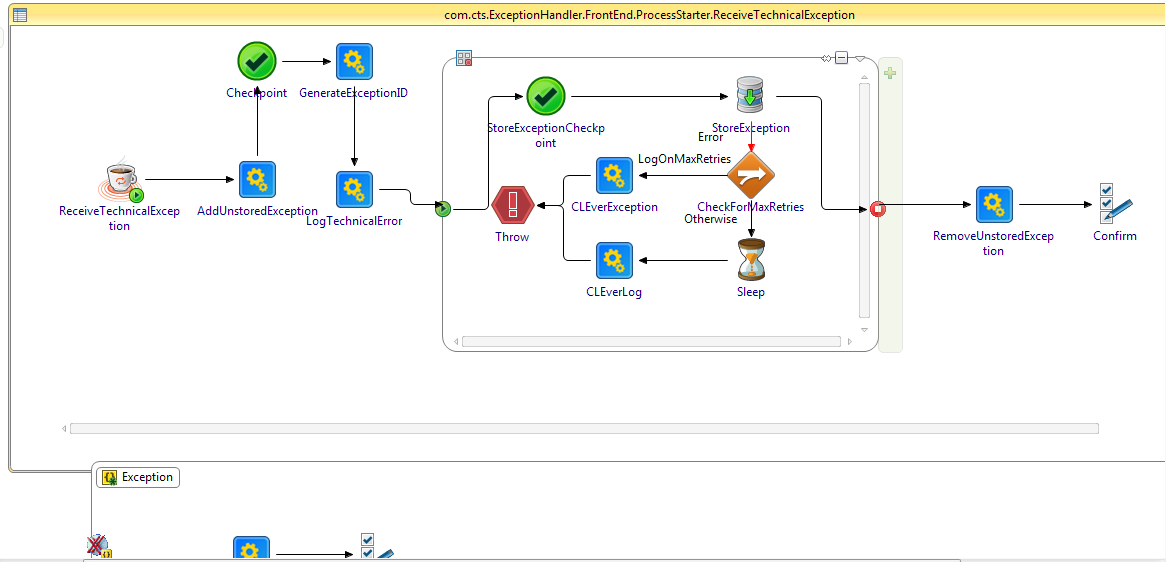
Exception Handling service:

* Basically the framework handles four types of exception.
* Technical Exception.
* Functional Exception.
* Check Dependency Exception.
* Release Dependency Exception.
* Technical Exception:-

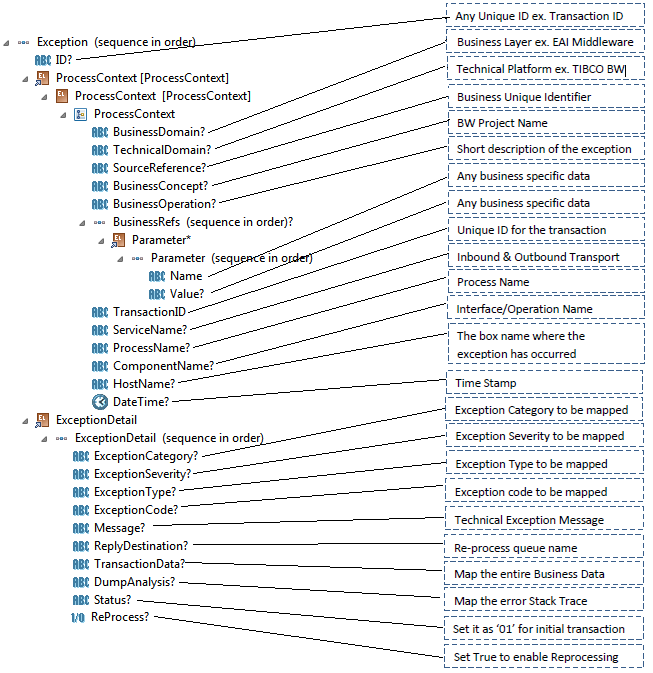
Technical Exception should be used to log any kind of anomalies occurred from a technical component in the project. Like any JMS, JDBC or HTTP exception form an internal or external source.



Process diagram to invoke technical exception



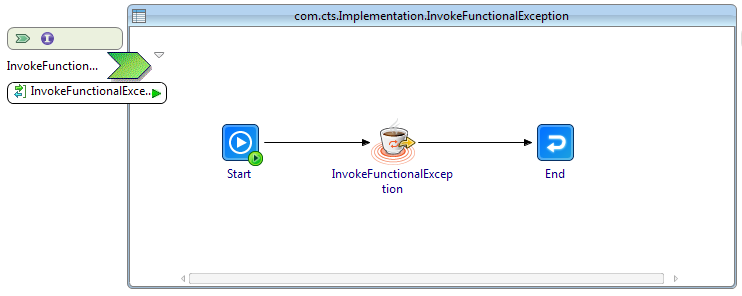
Process diagram of technical exception handler implementation



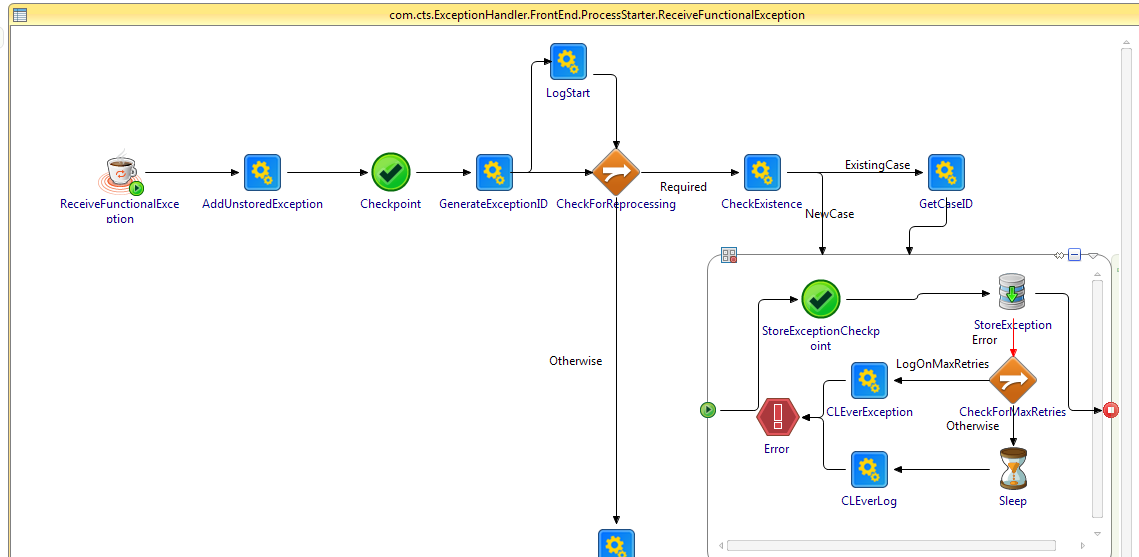
Schema mappings for technical exception

* Functional Exception:-

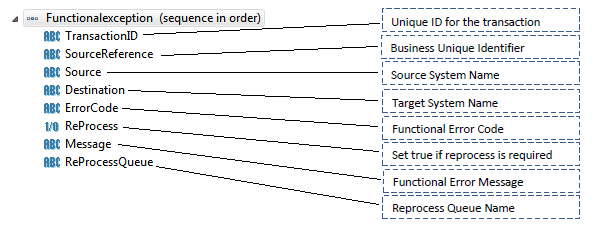
Functional Exception should be used to log any kind of business error responded from a third party system or any business validation failure inside the BusinessWorks implementation.



Process diagram to invoke functional exception



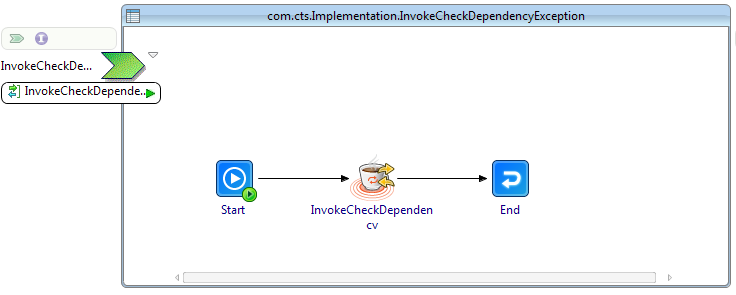
Process diagram of functional exception handler implementation



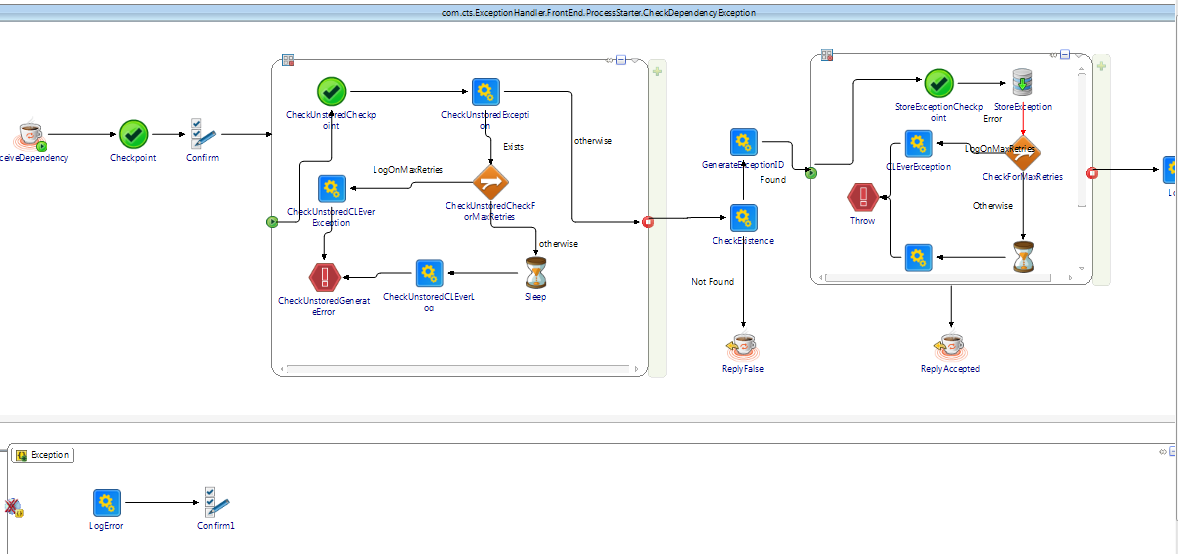
Schema mappings for functional exception

* Check Dependency Exception:-

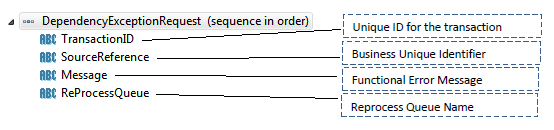
Check dependency exception can be used if there are dependency/relations between two transaction processed by the interface which interns enforce us to maintain a sequence of the transaction to be processed. Check dependency allows the interface to find any dependent transaction in process and keep the current transaction on hold.



Process diagram to invoke check dependency exception



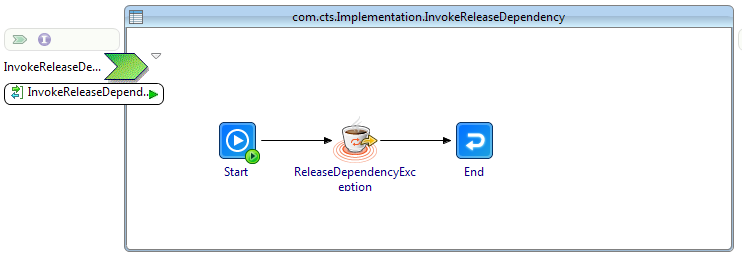
Process diagram of check dependency exception handler implementation



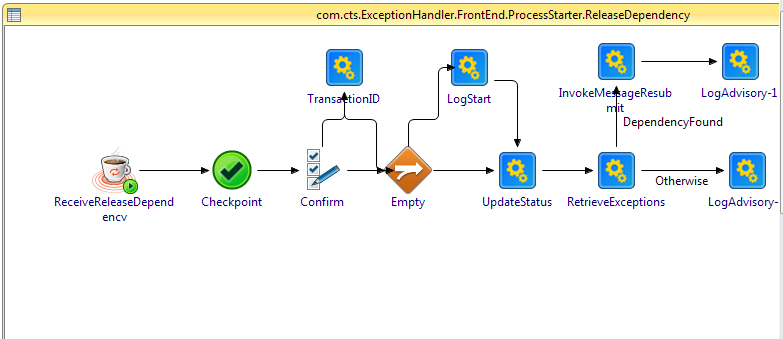
Schema mappings for check dependency exception

* Release Dependency Exception:-

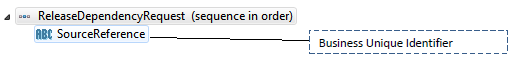
Release dependency exception can only be used if we implement check dependency exception in any interface. This is not a technical exception rather this helps us to remove the dependency from the system so that any further downstream transaction can be processed.



Process diagram to invoke release dependency exception



Process diagram of release dependency exception handler implementation



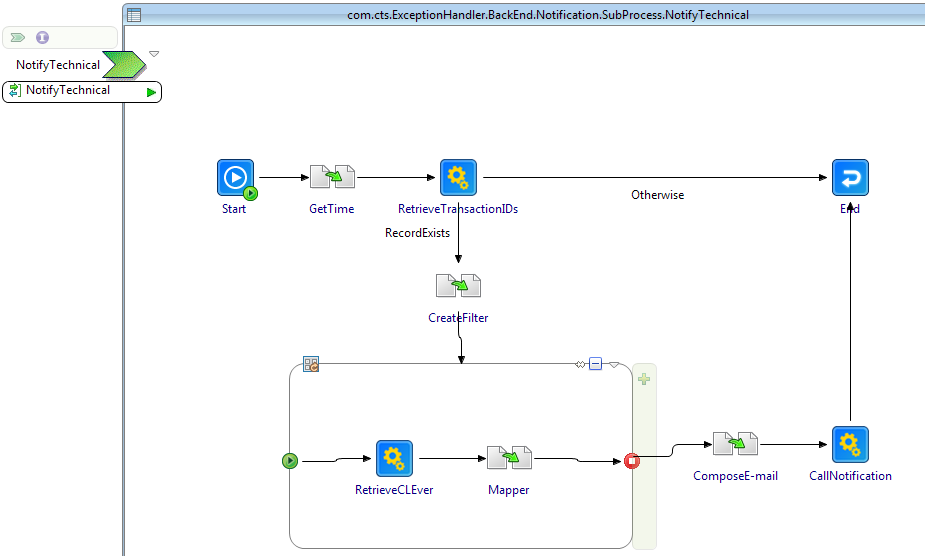
Schema mappings for release dependency exception

* Notification service:

Notification service enables to notify the end user about the exceptions that occurs. The notification is mainly send via email. There are two different ways in which notification service sends notification to the end user.

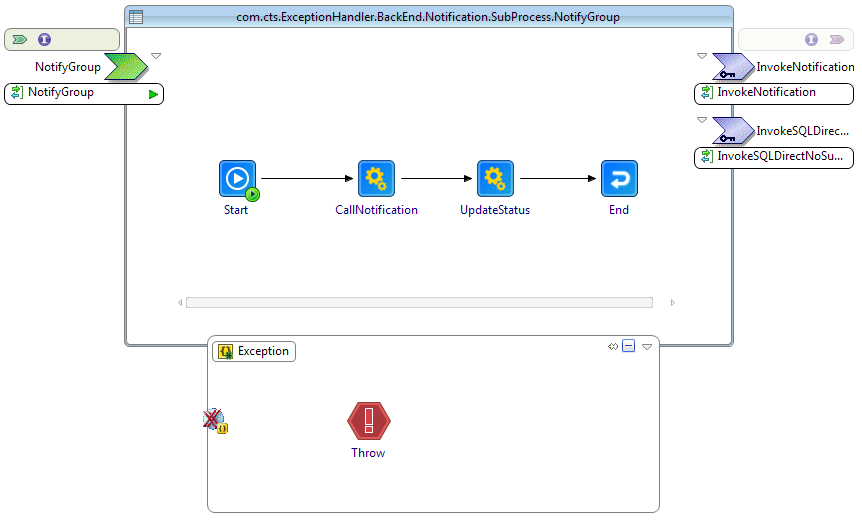
* One by one notification.
* Bulk Notification.
* One by one notification:-

In one by one notification, different notifications need to send to different recipients.



* Bulk notification:-

In bulk notification, different notifications need to send to a single recipient.



* Reprocessing:

Reprocessing enables the feature of retry mechanism. In downtime such as database down, web service down..etc. technical exception might need to retry every 1 hour.

