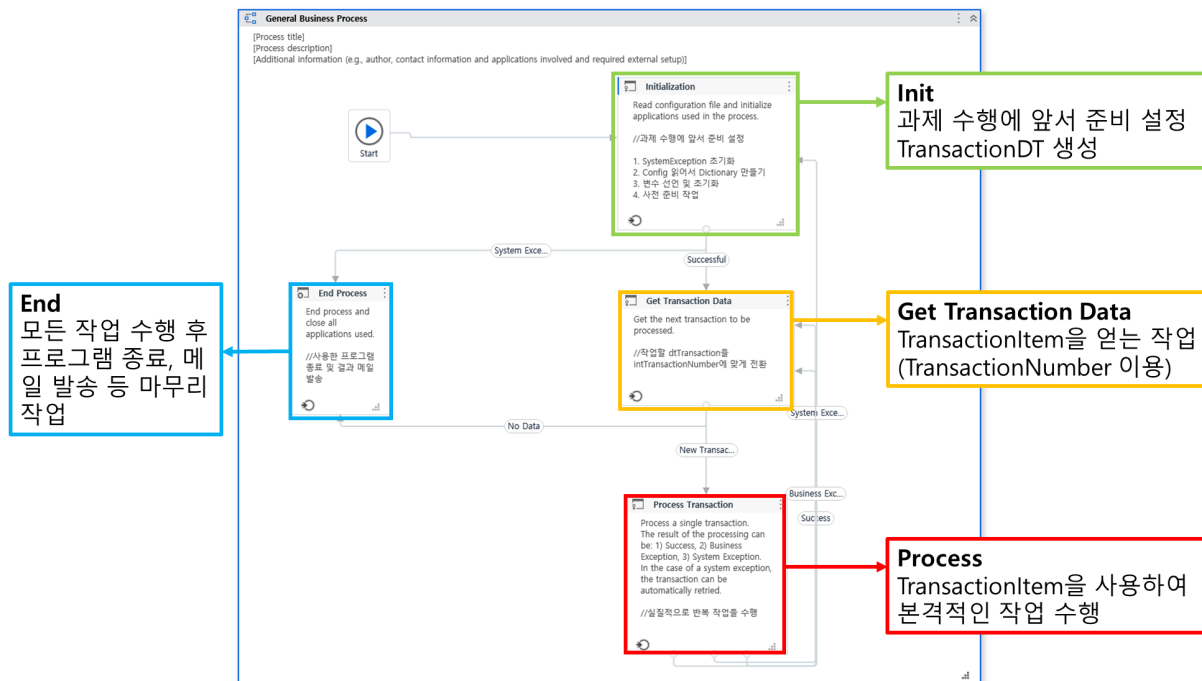


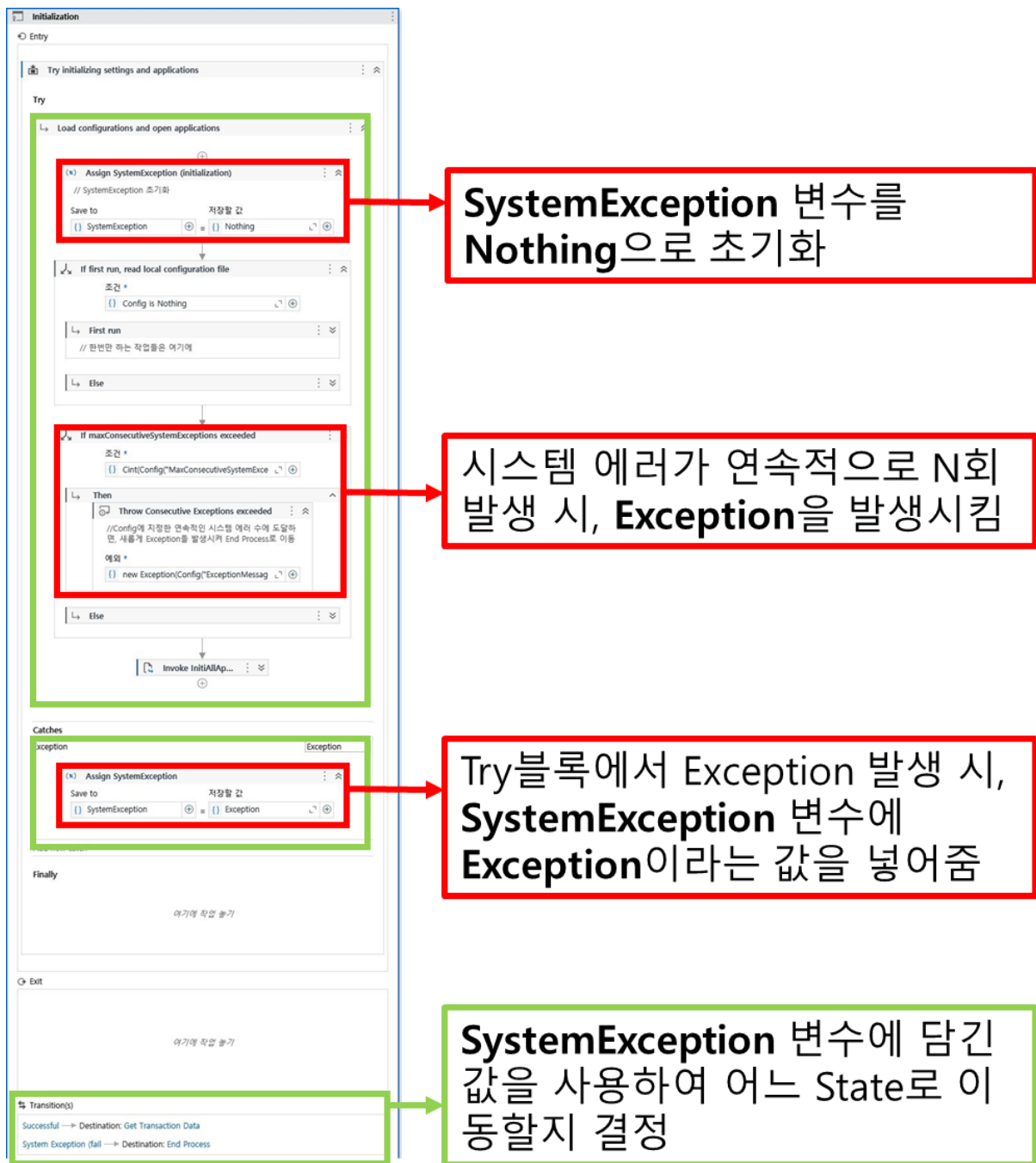


RE-Framework 개론

🍪 전체 구조



💚 Initialization

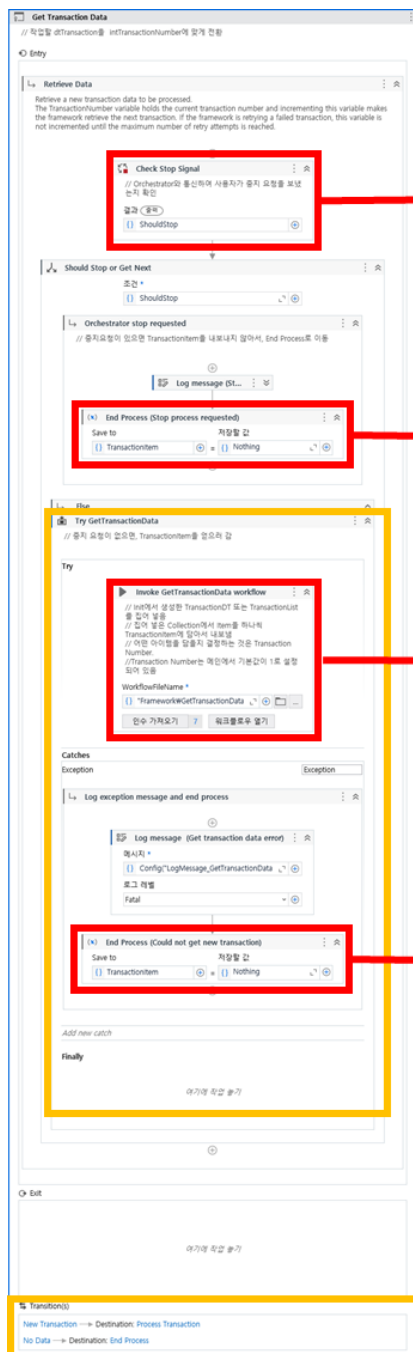


Init

- 과제 수행에 앞서 준비 과정을 설정하는 곳
- 주로 사용 변수 선언 및 `dtTransaction` 설정
- 작업할 사이트 Login도 이곳에서 수행. (때에 따라, 사이트 Login은 다른 곳에서 해도 됨)
- 정상적으로 작업한 경우 > `Get Transaction Data` 로 이동

- Error가 발생한 경우 > **End Process** 로 이동
- 그리고 나중에 **Process Transaction**에서 System Exception 발생하면 다시 돌아오는 곳이기도 함

♥ Get Transaction Data



오케스트레이터에서 Stop 신호가 있는지 여부

TransactionItem 변수에 **Nothing**을 줘서 No Data로 이동

TransactionDT와 **TransactionNumber**을 집어 넣고, 이를 사용하여 **TransactionItem**을 얻어오는 작업

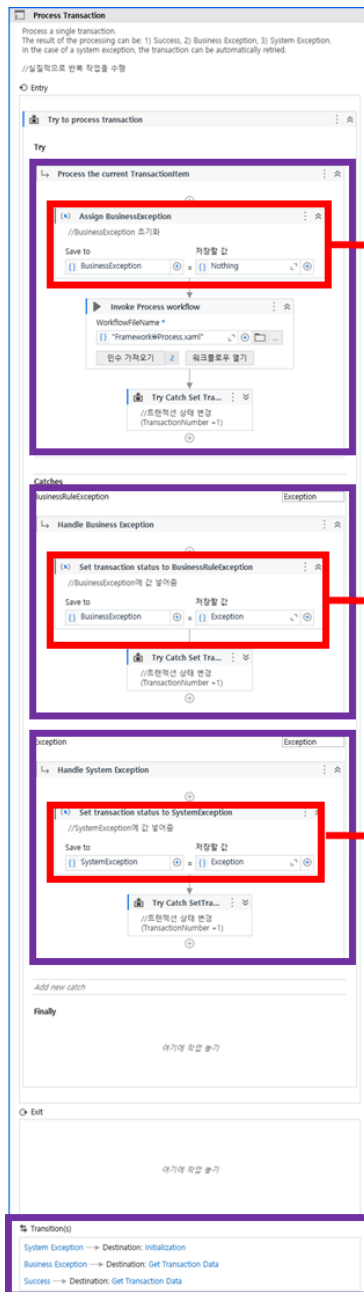
TransactionItem 변수에 **Nothing**을 줘서 No Data로 이동

TransactionItem 변수에 담긴 값을 사용하여 어느 State로 이동할지 결정

Get Transaction Data

- 작업할 `dtTransaction` 을 `intTransactionNumber` 에 맞게 `drTransactionItem` 으로 변환해주는 곳
- 해당 xaml 파일은 가급적 수정하지 않는 것이 규칙
- 정상 작업한 경우
 - 작업할 Data가 있으면(`TransactionItem<>Nothing`) > `Process Transaction` 로 이동
 - 작업할 Data가 없으면(`TransactionItem=Nothing`) > `End Process` 로 이동
- Error 발생한 경우
 - 실패한 Log를 남기고 `End Process` 로 이동
- 그리고 **Process Transaction**에서 아래의 경우에 돌아오는 곳이기도 함
 - 정상적으로 작업한 경우 (**Success**)
 - Business Exception이 발생한 경우 (**Business Exception**)

Process Transaction



BusinessException 변수를 **Nothing**으로 초기화

Try블록에서 Exception 발생 시, **BusinessException** 변수에 **Exception**이라는 값을 넣어줌

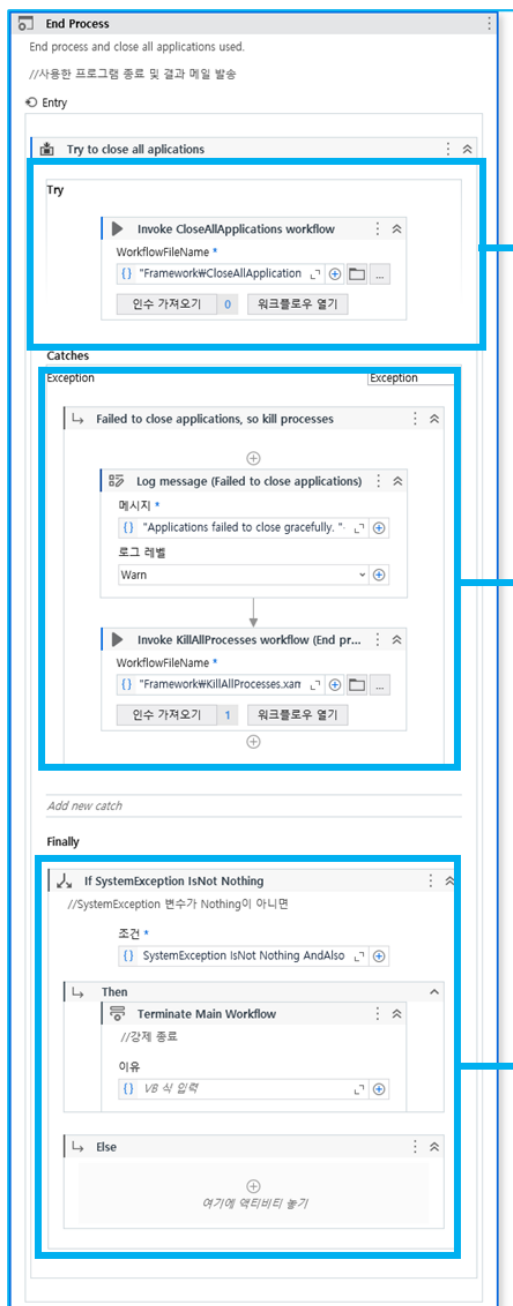
Try블록에서 Exception 발생 시, **SystemException** 변수에 **Exception**이라는 값을 넣어줌

SystemException 변수와 **BusinessException** 변수에 담긴 값을 사용하여 어느 State로 이동할지 결정

Process Transaction

- 실질적으로 반복 작업을 수행하는 곳
- 정상 작업한 경우(Success) > **Get Transaction Data** 이동
- Error 발생한 경우
 - System Exception > **Initialization** 이동
 - Business Exception > **Get Transaction Data** 이동

♥ End Process



End Process에서 실질적으로 할
작업
(Ex. 프로그램 종료, 메일 발송
등)

End Process의 Try에서
Exception이 발생하면 하는 작
업

SystemException 변수 값으로
강제 종료여부 결정

End Process

- 모든 작업이 완료되어 마지막으로 오는 곳
- System Exception이 발생했는지 여부에 따라 작업을 다르게 구현 (Try 블록 안에서)

- 성공 시 : 엑셀 기록, 성공 메일 발송 등
- 실패 시 : 실패 메일 발송 등
- **SystemException**이라는 변수에 담긴 값으로 강제 종료할지 여부 결정 (Finally 블록)