ADEPTIA

443 North Clark St, Suite 350
Chicago, IL 60654
Phone: (312) 229-1727

Exception Handling in Process Designer

User Guide

Introduction:

Exception handling is the process of responding to the occurrence, during computation, of *exceptions* – anomalous or exceptional situations requiring special processing – often changing the normal flow of program execution. It is provided by specialized programming language constructs or computer hardware mechanisms.

In **Adeptia Suite** exception can occur at any entity due to various reasons like **File Source Activity** fails to retrieve the requested file or **Data Mapper Activity** fails due to parsing of invalid data, so to recover from these exceptions or error and to go with alternative flow of activities you have two ways to handle it which are as follows:

1. By creating exception handler script at Activity or global level.
2. By using error Intermediate event at activity level.

These two methods work independently and preferred according to the situation. Suppose if exception occurs at any activity and you only want to log more diagnostic message about the activity on which exception comes and about Process Flow containing that activity then exception handler script is used, but if user want to follow alternative flow of activities i.e. user want to retrieve resource located on another location then error Intermediate event is used.

In case if user have defined both exception handler script and error Intermediate event for an activity then first exception handler script is invoked and its functioning will occur, after that process flow follows the path headed by error Intermediate event instead of aborting the process flow.

But if somehow exception handler script fails due to Syntax or runtime exception then Process Flow will abort and follows the path after activity on which exception occurred by skipping all further entity reaching to end event.

Basically exception handler Script is used to run any custom java code to perform some functioning whenever any exception occurs, whereas error Intermediate event is used to divert flow to another path with set of arranged activities headed by it to perform task after exception occurs such as to retrieve file from another location or to update another table.

Creating Exception Handler Script

Exception Handler scripts are basically Java code which is invoked if any exception or error occurred during execution of a process flow. There are three types of Exception Handler Scripts:

- Service Exception
- Process Flow Exception
- Invalid Data Exception

These Exception Handler Scripts can be created at a global level (for all the activities in the Process Flow) or at the activity level (for a specific activity). If an exception or errors during the execution of a process flow, the activity level exception handler script is invoked first. If the activity level exception handler script is not created for that activity, then only the global level exception handler script is invoked.

These Exception Handlers are invoked at different stages of the process flow. These are depicted in the table below.

Table 1: Exception Handlers in a Process Flow

Service Exception	This exception handler script is invoked when any error related to service (activity) occurs. It can be defined for activity as well as global level .For example: File not found, Stream closed etc.
Process Flow Exception	This exception handler script is invoked when any error occurs at Process Flow level. The Exception can only be declared at global level. For example: JMX not found, Repository not found etc.
Invalid Data Exception	This exception handler script is invoked when any error related to processing of data occurs. It can be defined for activity as well as global level. For example: Incorrect record format etc.

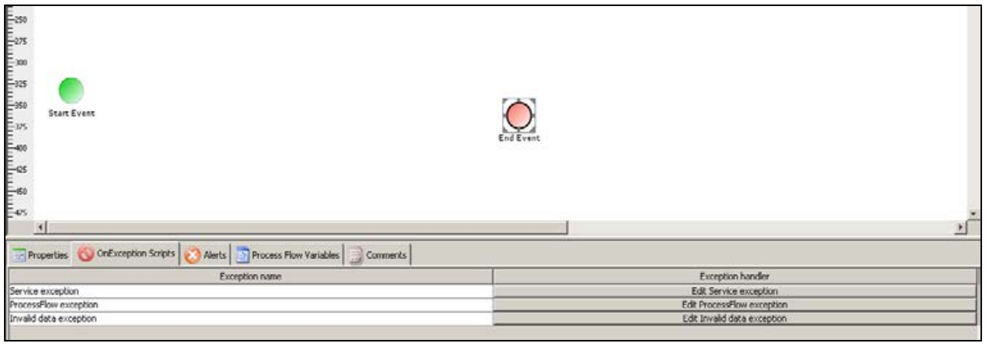
If you have defined Service or Invalid Data exception handler script at both level i.e. at Activity as well as global level then only activity level script will invoked whenever exception occur on that activity. It signifies that global level script is invoked only when there is no activity level script is defined.

Process Flow Exception handler script is invoked whenever process flow aborts due to some reason. Let's say if an activity fails but you have already attached error intermediate event then it divert path to other activity and Process Flow doesn't abort and will not invoke its handler script.

Creating Global Exception Handler Script

Steps to create a Process Flow (Global) Exception Handler Script in Process Designer

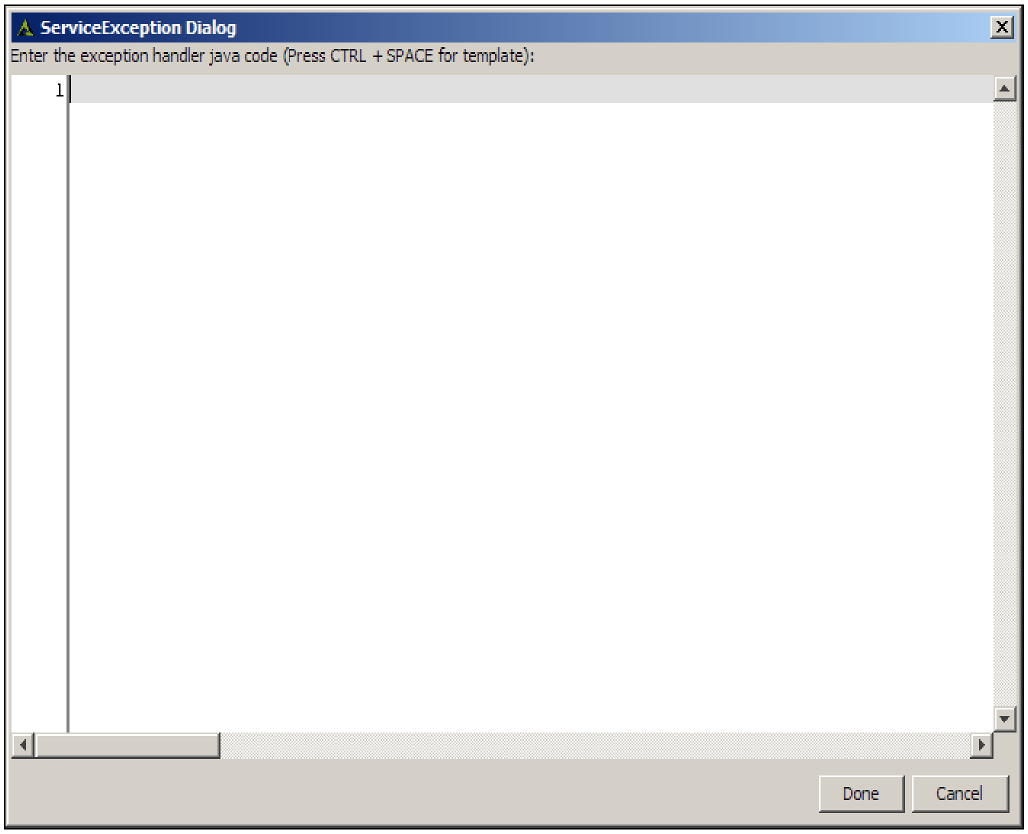
1. Click the **On Exception Scripts** tab ( ) in the Bottom Pane. The **Create Exception Handler** screen is displayed (see Figure 1).



Exception name	Exception handler
Service exception	Edit Service exception
ProcessFlow exception	Edit ProcessFlow exception
Invalid data exception	Edit Invalid data exception

Figure 1: Creating Exception Handler Script

2. Click **Edit Service Exception**. The **Service Exception Dialog** window is displayed (see Figure 2).



Enter the exception handler java code (Press CTRL + SPACE for template):

Figure 2: Service Exception Dialog Box

You can use **<CTRL>+<Space Bar>** to view pre-defined template of Java Code, which can be used in creating Java Condition. You can select any of them and edit it according to your requirement.

If you want to create the process Flow Exception Handler or Invalid Data Handler script, click **Edit Process Flow Exception** or **Edit Invalid Data Exception** buttons respectively.

3. Enter the Java code in the **Service Exception** dialog box and click **Done** button.

Creating Activity Exception Handler Script

Steps to create an Activity Exception Handler Script in Process Designer

1. Right-click the activity in the Graph Canvas and select **Service Exception Dialog**. The Service Exception Dialog Box is displayed (see Figure 121).
2. Enter the Java code in the **Service Exception** field and click **Done** button.

You can use **<CTRL>+<Space Bar>** to view examples of Java Condition. **<CTRL>+<Space Bar>** show lists of example. You can select any of them and edit it according to your requirement.

If you want to create Invalid Data Handler script, right-click the activity and select **Invalid Data Exception Dialog**.

There are four implicit objects available in exception handler scripts which are as follows:-

1. **exception**: - It is object of Exception class so you can invoke all methods of this class on it. For example **getMessage ()** returns the detail message string of this throwable and **printStackTrace** () method prints a stack trace for this Throwable object on the error output stream i.e. on Kernel or Process Flow logs.

   Enter the script for the information you want to know if exception occurs , e.g. If you want to gets information regarding cause of exception then use following script:-
   String exceptionMessage = exception.getMessage ();
   System.out.println (exceptionMessage);

2. **context**:- This object is used to access the transaction's context or to get or set the context variable (to know more about context variable refer to section CREATING CONTEXT VARIABLE of Developer Guide) e.g. if exception occurs you want to get or set the variable's value then use the following script:-
   context.get ("Name_of_Varible"):-used for getting the value of variable.
   context.set ("Name_of_Variable", Value_of_variable):-used for setting the value of variable.

   Suppose you want to know the Transaction ID whenever exception occurs then use following code:-
   String transId = context.get ("TransactionPID");
   System.out.println ("Process Flow Transaction Id is" +transId);
   It prints the transaction Id on Kernel logs.

3. **service: -** This object is used to get the information about the activity e.g. if you want to know about the name of activity wherever exception occurs then following script will be used.
   String activityName = service.getName ();

System.out.println ("Name of activity on which exception occurs is "+activityName);
It gives the name of activity on which exception occurs so that user can use it for notification.

4. **log**: - This object is used to show information on Process Flow Log so that user can get the customized information regarding the exception or error occurred. As there are three modes of Logging Level **(DEBUG, INFO & ERROR)** therefore you can get information according to log level set in process flow.

   log.debug ("customized message"):- It shows the message in Diagnostics link of Process Flow Log whenever Process Flow runs as DEBUG logging level.

   log.info ("customized message"):- It shows the message in Diagnostics link of Process Flow Log whenever Process Flow runs as INFO logging level.

   log.error ("customized message"):- It shows the message in Diagnostics link of Process Flow Log whenever Process Flow runs as ERROR logging level.

In case if exception or error occurred by exception handler script then exception will be thrown and process flow would be aborted, so while creating exception handler script please make sure that you are writing correct syntax and exception free script otherwise there would not be any constructive use of it as it will compile and execute while process flow is running.

USING ERROR INTERMEDIATE EVENT

Error Intermediate Event is used to redirect Process Flow execution to an alternate path in case of failure of any activity during process flow execution. To understand the use of Error Intermediate Event considers the following scenario:

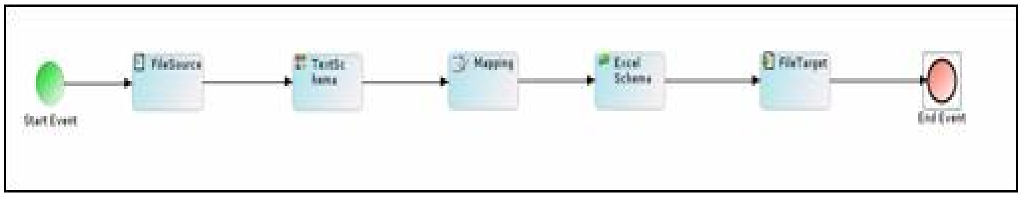


Figure 3: Scenario

In this process flow, data from a text file is converted to an excel file using schema and mapping activities. Now if the Mapping activity is critical for your business, you may want be notified, if mapping activity fails during process flow execution.

To handle this situation you can attach **Error Intermediate Event** with mapping activity so that, in case mapping activity fails, a notification activity is executed and sends an email to the specified email address. Use of Error Intermediate Event is displayed in the Figure 4

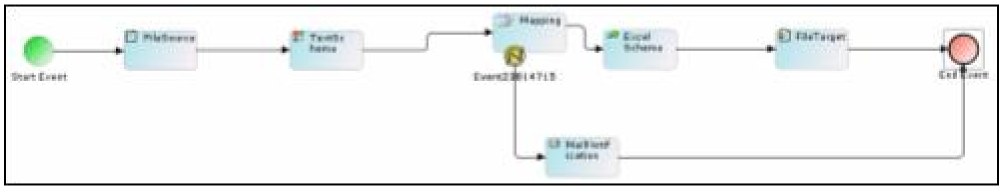


Figure 4: Use Error Intermediate Event

Steps to use Error Intermediate Event in Process Designer

1. Right-click the activity with which you want to attach the intermediate event and select **Add Intermediate Event** option (see Figure 5).

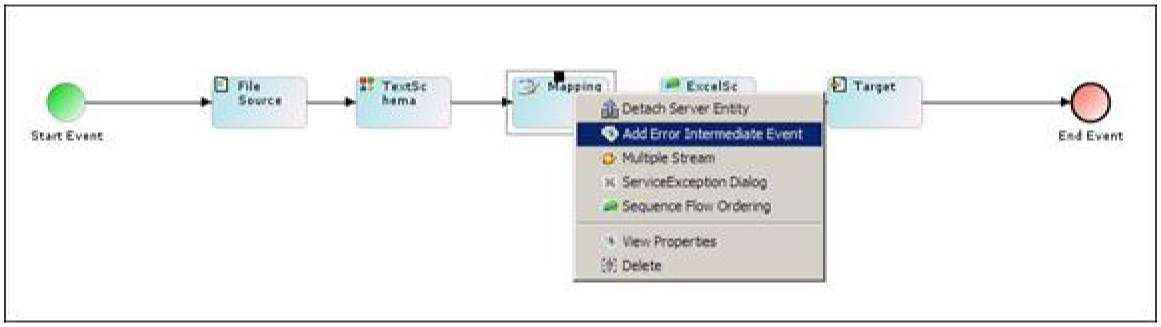


Figure 5: Select Add Intermediate Event

2. This attaches the Intermediate Event to the selected activity (see Figure 6).

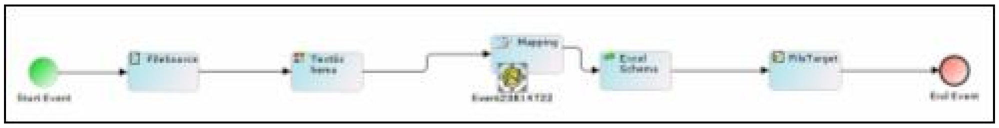


Figure 6: Error Immediate Event Attached

3. Drag another activity, which needs to be executed in case of failure of Mapping activity, to the Graph Canvas Area (see Figure 7).

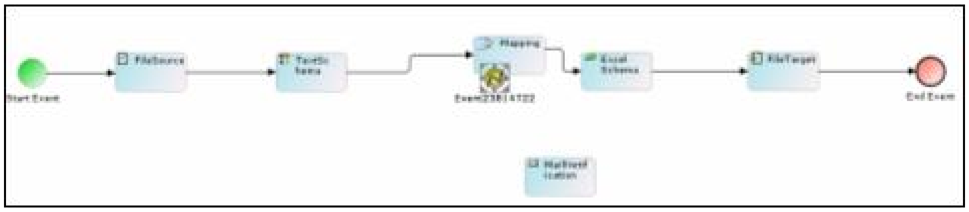


Figure 7: Drag Another Activity

4. Connect the Error Intermediate Event to Mail Notification activity and then Mail Notification activity to End Event (see Figure 8).

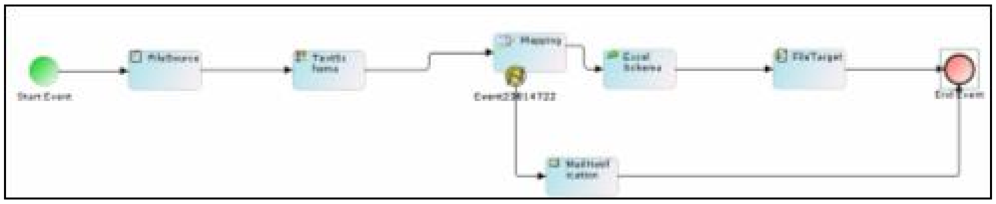


Figure 8: Connect Activities

While connecting Error Intermediate Event with Mail Notification activity, please ensure that you are connecting Error Event with Mail Notification not the Mapping activity with Mail Notification.