

Transaction Code Flow

The aim of this document is for the application SME to capture **representative** transactional code flows snippets between the entry and end points of the application. It is not intended to capture flows between all the individual instances of entry/end points (explained below) but rather to capture a representative flow through each of the technologies depicted in the Architecture Context Diagram. This document is used by the CAST Administrator during the analysis configuration workflow stage ([eBlue Print - Analysis Configuration and Execution](#)).

Download the document template from here [Transactional_Code_Flows_appName.docx](#). The template contains an example representative code flow, shown below, of what is required. If an equivalent document already exists which shows what is required then provide that instead of creating a new document.

Typical example where an application has the following technology interactions between entry and end points then one representative flow is required for each of the following:

1. HTML /JSP (entry points) -> Struts Action -> JAVA -> Hibernate -> Database (end points)
2. HTML/JSP (entry points) -> Struts Action -> JAVA -> Spring Beans -> Hibernate -> Database (end points)
3. HTML /JSP (entry points) -> Struts Action -> JAVA (end points) -> MQ
4. MQ -> MDB (entry points) -> JAVA -> Spring Beans -> Hibernate -> Database

The representative code flows must also include and show, where applicable, how the application :

1. interacts with the application databases and external databases
2. database is inserted/updated/deleted/queried by external applications or SQL Scripts
3. how the application interact with external applications (non-database) e.g. sends and/or receive file feeds, ETL Tools, JMS, MQ, MDB, etc
4. uses batch processes
5. interacts with webservices (consumer/provider)
6. uses configuration code and mapping of any Custom XML files e.g. custom xml mapping files that define relationship such as stored procedure name with corresponding variable name within the code

What are Entry and End Points?

In order to compute the transaction Function Points, the transactions need to be identified from the entry points to the transaction end points. An application entry point identifies a resource that is an access point to an application such as Web service called by another application, webpages, batch files, JCL, COBOL Program, Interfaces, classes which receive external messages e.g. MQ. An application end point is a user interface end point or a service end point such as database tables, files operations, call to another system by websevice, classes which send external messages e.g. MQ, OutputStream.

The template shows the following example :

Flow starts with default.aspx and then all the flows moves depending upon the querystring value of the URL

1) Go to masterconfig.js to get all the querystring values i.e.

```
"editmasterconfig.js" : "./default.html?uid=editmasterconfig.js",
"requestSchema" : "./default.html?uid=requestSchema",
"changeLimits" : "./default.html?uid=changeLimits",
"contactinformation" : "./default.html?uid=contactinformation",
"bill" : "./default.html?uid=bill",
"invoice" : "./default.html?uid=invoice",
```

2) Then go to corresponding JS file for particular HTML page and look for KEYWORD == "CONFIGSOURCES" to get the name channel service method name i.e.

```
refDataUtil : null,

configSources : [ "group/.../MasterConfig.js", "group/.../MasterText.js" ],
_rtrv...inDetlRef : "retrieve...erDetails",
_rtrv...lEndPoint : null,

_rtrv...lRef : "...tConnDtl",
_rtrv...EndPoint : null,

/*
```

3) From here go to the masterconfig.js to get the actual method name of service class

```
),
"customer": {
  "retrieveCustomerDetails": {
    "requestSchema": ".../acct/schema/...quest.json",
    "responseSchema": ".../acct/schema/...sponse.json",
    "url": global_api...textPath + "/channel/proxy/cus...aSvc/rtrvCustPrsnDetl",
    // "url": ".../acct/schema/s...tl.json",
    "cache": "page",
    // "cache": "...",
    // "cacheTime": "... seconds",
    // "cacheTimeout": 900
  },
  /*rtrvCustPrsnDetlForMasterconfig.js*/
```

4) Go to service implementation of channel service method

```
@Override
@RequestMapping(value = "/rtrvCustPrsnDetl", method = RequestMethod.GET)
@ResponseBody
public RtrvCustPrsnDetlResponse rtrvCustPrsnDetl(
    @Context(context = CustomerContext.IN) CustomerContext context,
    final Request request) throws BusinessException {
    try {
        CustMgmtImpl.LOGGER.debug("inside rtrvCustPrsnDetl method of CustMgmtImpl");
        com.bankgroup.bussvc.ipsgmt.custmgtimpl.RtrvCustPrsnDetlRequest busRequest = new com.bankgroup.bussvc.ipsgmt.custmgtimpl.RtrvCustPrsnDetlRequest();
        CustomerIdentifier identifier = new CustomerIdentifier();
        identifier.setCustomerId(context.getId());
        busRequest.setIdentifier(identifier);

        com.bankgroup.bussvc.ipsgmt.custmgtimpl.RtrvCustPrsnDetlResponse busResponse = this.custMgmtImpl.rtrvCustPrsnDetl(busRequest);

        RtrvCustPrsnDetlResponse rtrvCustPrsnDetlResponse = new RtrvCustPrsnDetlResponse();
        this.mapper.map(busResponse, rtrvCustPrsnDetlResponse);

        return rtrvCustPrsnDetlResponse;
    }
}
```

5) Then this method call the business service method i.e.

```
headers = (HashMap<String, Object>) this.contentType().get("Content-Type").get("application/json");
// method called for logging
this.logger.info("Request Headers: " + headers);

response = this.customerService.getCustomerDetails(request, headers);
} catch (Exception mEx) {
    if (mEx.getCause() instanceof BusinessException) {
        throw (BusinessException) mEx.getCause();
    } else if (mEx.getCause() instanceof com.liferay.portal.kernel.bean.BusinessException) {
        BusinessException busException = new BusinessException(
```