**Batch Feed Sequencing**

**Salient points:**

1. There are a total 7 batch feeds that are specific to Max and Access. These are Customer, CustomerXref , Division, Entitlement, PriceBook, ItemBranch and Uom. Each of these batch feeds will be configured as a service specif to Max/Access. The service names and the repository(service group) that these services fall under are as given in the attached document:

The other batch feeds that are not specific to Max/Access are Item, Attribute, Category and Master UOM. These services also find a mention in the table above.

Note: The sequencng logic will only be applied to the services that are specific to either Max/Access.

2. There will be a total of 4 Integration servers that will be used for the batch feeds, namely:

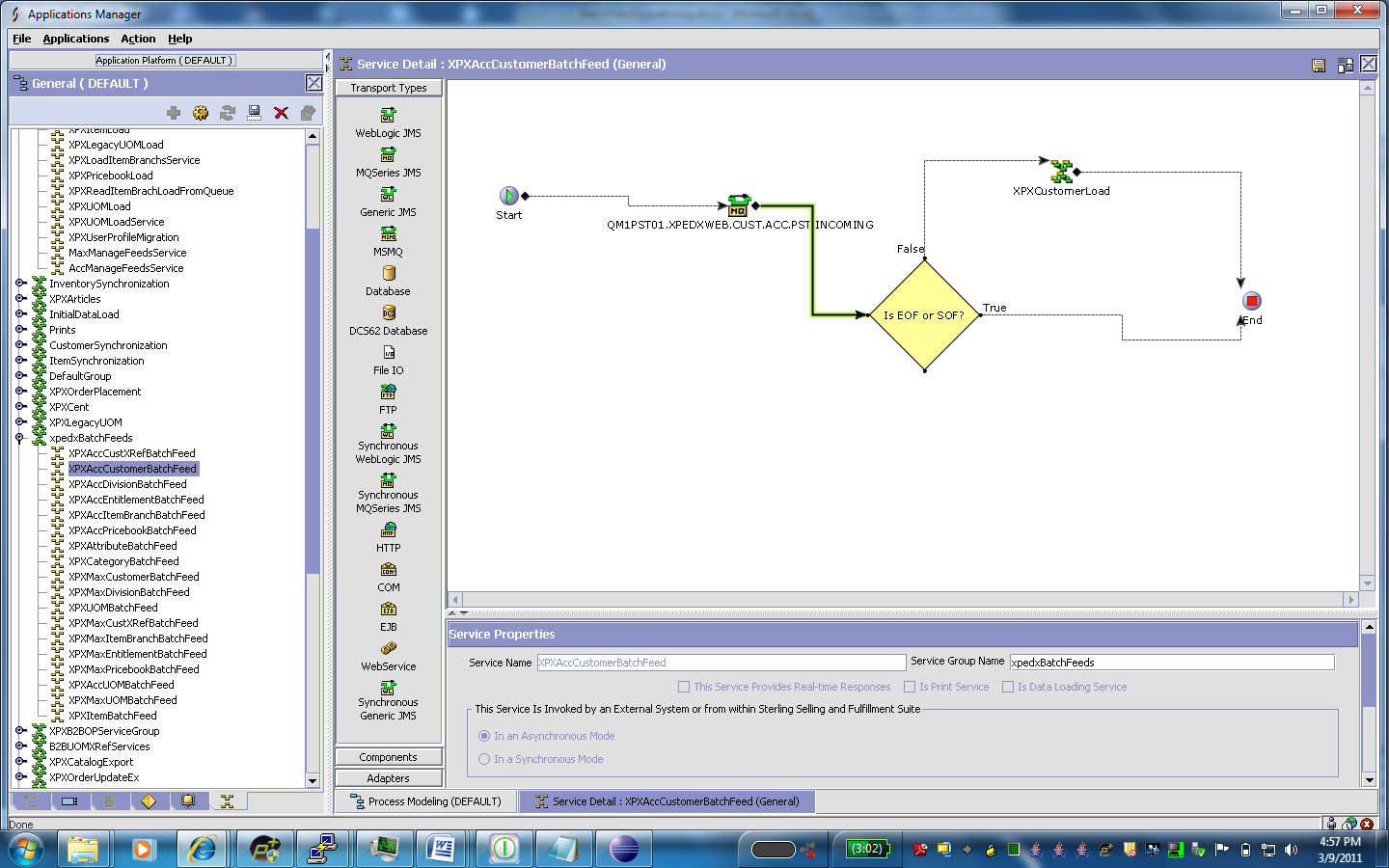
a) AccBatchFeedServer: For all the 7 Access batch feeds

b) MaxBatchFeedServer: For all the 7 Max batch feeds

c) XPXDataFeedMQServer: For Item, Attribute and Catalog batch feeds

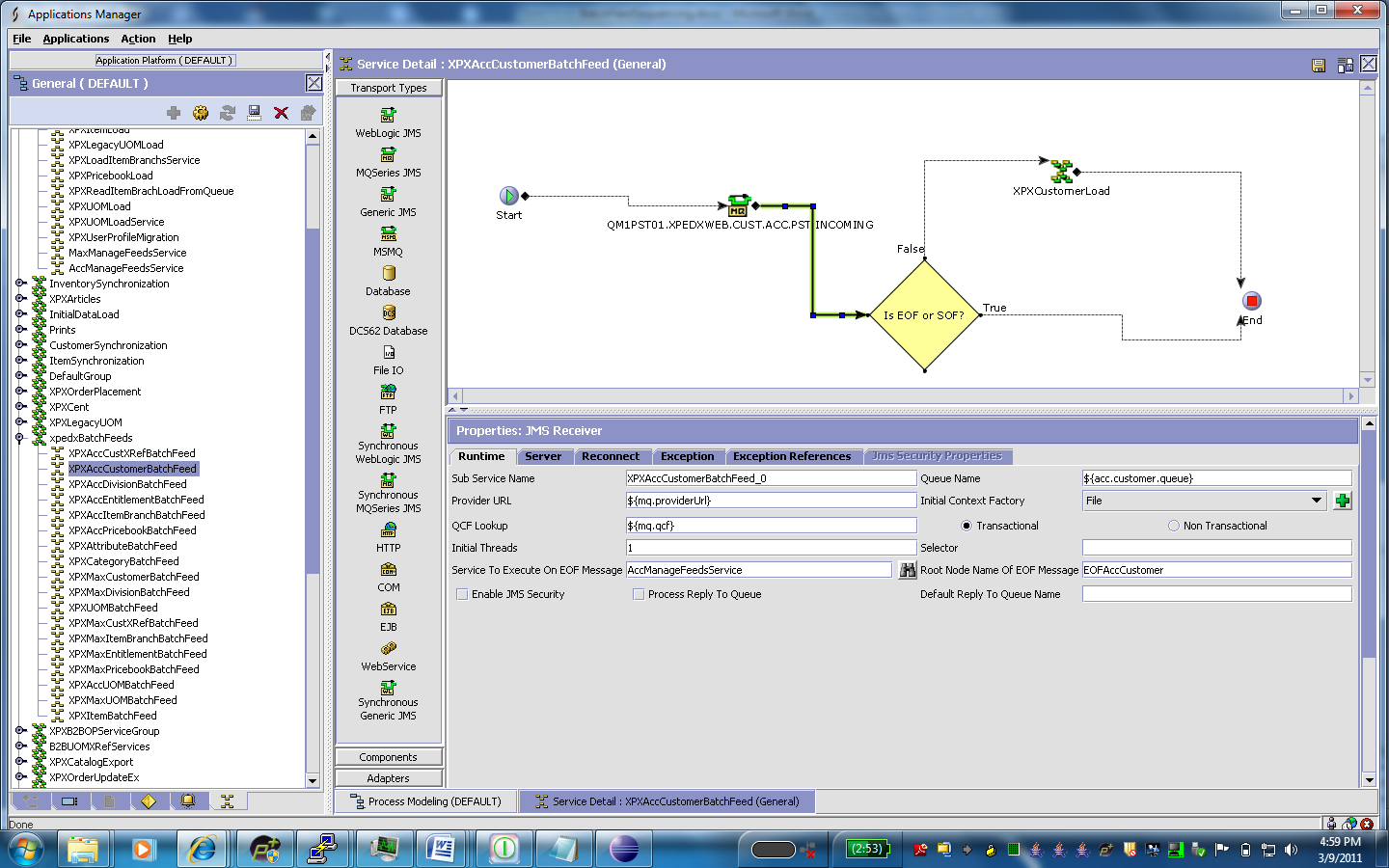
d) xpedxuomserver: For the Master UOM batch feed.

3. An example of a batch feed(the ones specific to Max/Access) that is configured is shown below with a brief description following it:



Each batch feed service will consist of a MQ queue follwoed by a condition to check if the message is an SOF/EOF message and then the service which actually loads the data.

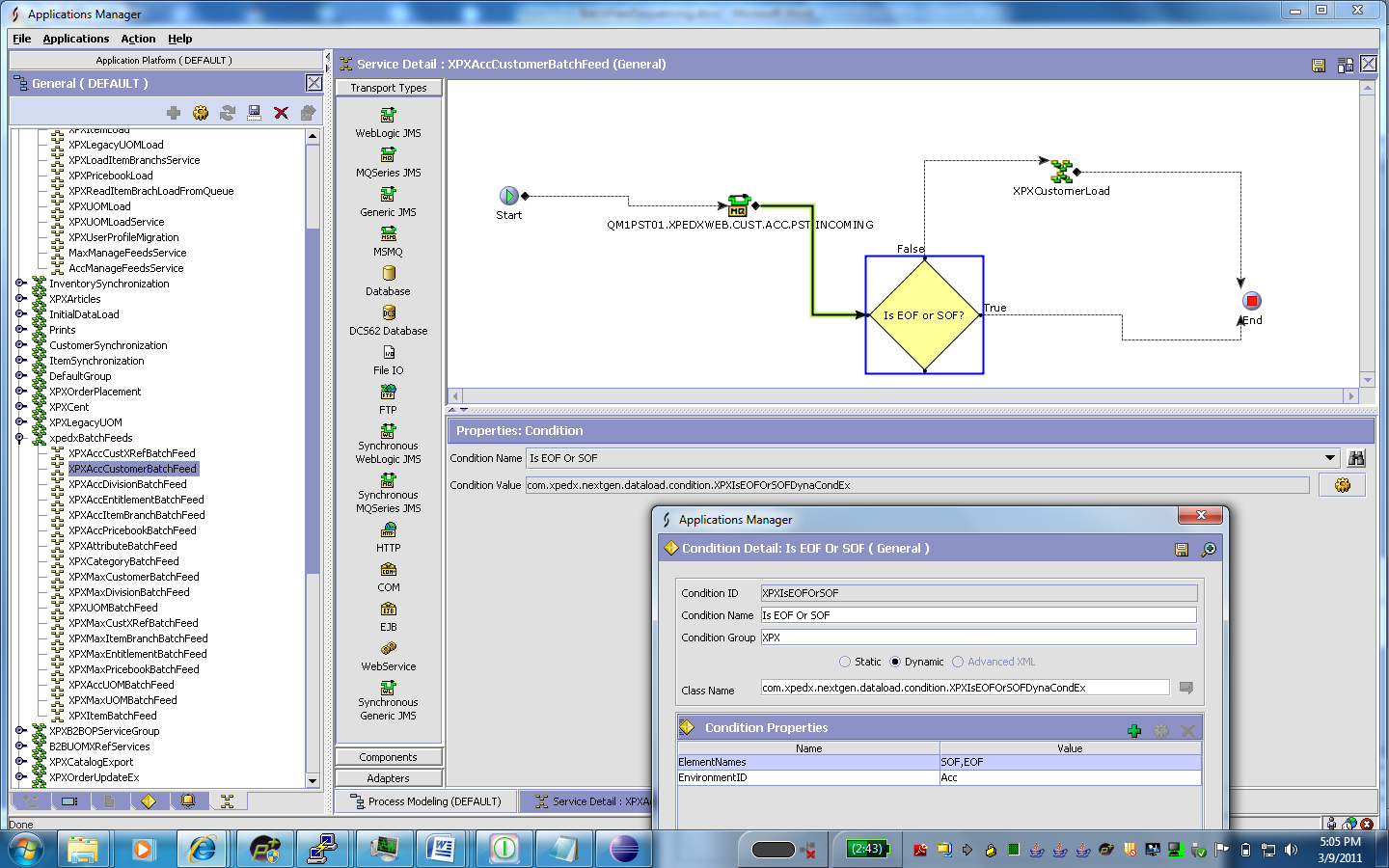
An example of the MQ configured(for the batch feed shown above) is as follows:



The queue details like provider url, queue name, etc. will be variables which will be read off the customer\_overrrides.properties file. Eg. For Queue name ${acc.customer.queue}, the corresponding customer\_overrides entry is yfs.acc.customer.queue=QM1PST01.XPEDXWEB.CUST.ACC.PST.INCOMING

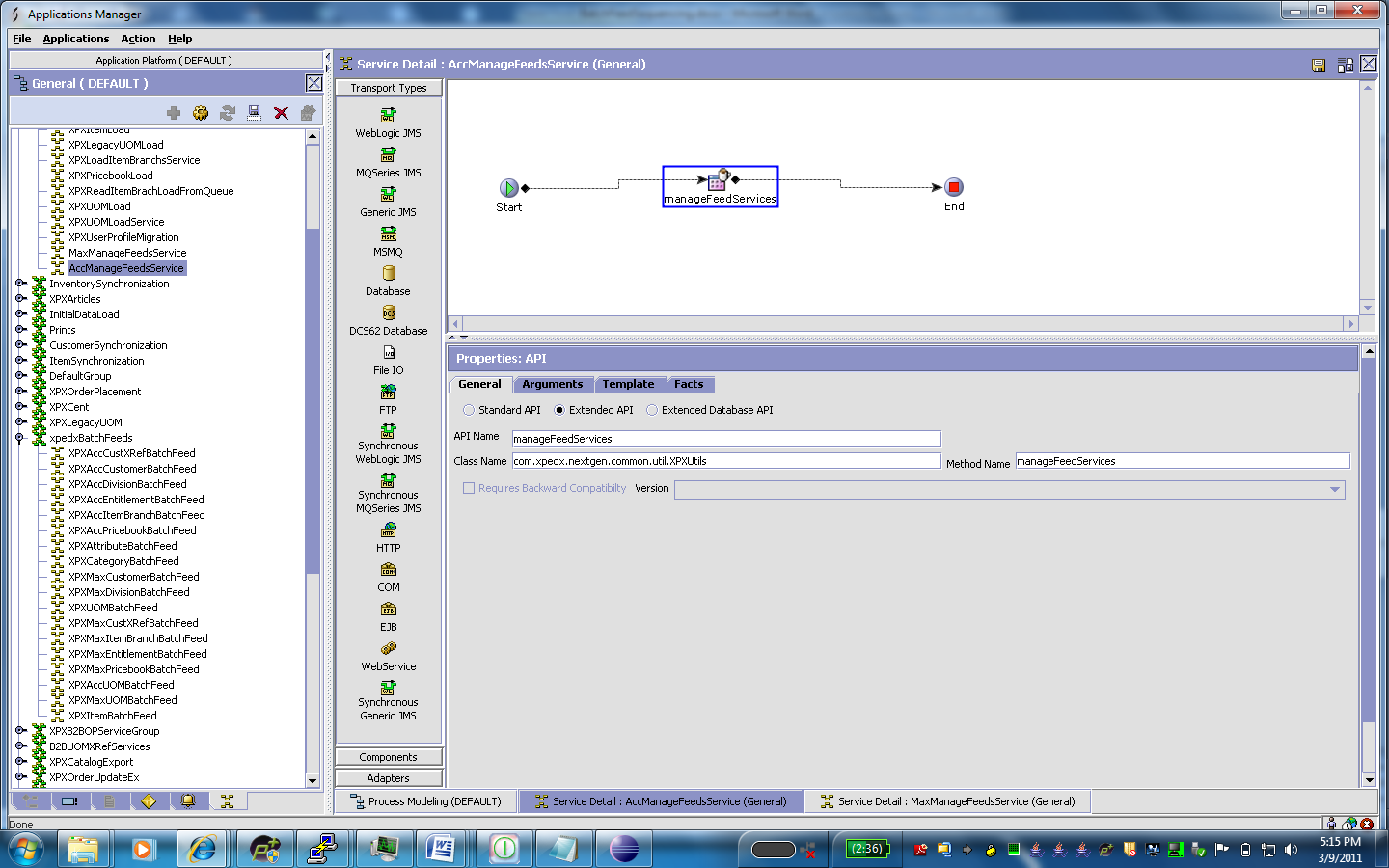
The integration server to which this queue is linked to is available on the next tab named ‘Server’.

An example of the condition which checks for a SOF message is:



This will be having two arguments where one of them refers to the environment in which this service falls i.e Access/Max.

4. The batch feed service directs the SOF/EOF message to a service that is configured alike for Access/Max where the only differences are in the arguments for each of them. This service is used to log the incoming feed message(SOF/EOF) into CENT(Common Error Notification Tool), any errors into CENT, resume/suspend the services depending on the incoming feed xml and the corresponding service and modify the cache to enable/disable the DB objects depending on the feed that comes in. The service that’s configured for Access is AccManageFeedsService and for Max its MaxManageFeedsService. A snapshot of AccManageFeedsService is as follows:



There are a total of 41 arguments for Acc/Max MangeFeedsService. The arguments are already configured in MC for the nw setup.A brief description of the arguments are as given below. If the values are subject to change at a later point of time, please change it accordingly.

1) **Integration Server Name**: This is specific to Access/Max and refers to the server linked to the queue for that feed.

2) **SOFElementNames**: The possible values the root node name of a SOF message can take.

3) **EOF element names liek EOFCustXref,EOFCustomer,EOFItemBranch, etc**.: These are the possible values that the root node name of an EOF message can take. The message that comes into the queue will always have the root node name as EOF, but in order to distinguish between the different services, we have configured an unique root node name in each service.

4) **Arguments that have the feedname suffixed by the stringTransType, eg CustomerTransType, PriceBookTransType, etc**.: These are used to retrieve the attribute the CENT rrequires while logging an entry for an incoming SOF/EOF message. The possible value are as follows:

|  |  |
| --- | --- |
| **Attribute Name** | **Attribute Value** |
| EntitlementTransType | Ent-B |
| PriceBookTransType | PB-B |
| CustomerTransType | Cust-B |
| CustomerXrefTransType | Custxref-B |
| DivisionTransType | Div-B |
| ItemBranchTransType | ItemBranch-B |
| UomTransType | UOM-B |

5. **The arguments which have the same name as the feedname in a SOF message or a YantraMessageGroupID in an EOF message**: These will have the corresponding subservice names that are configured for the queues in the corresponding services. These are used as parameters during invocation of modifyServer API which requires these values to resume/suspend a service. Example: ArgumentName: Customer ArgumentValue:XPXAccCustomerBatchFeed\_0

6. **The ListOfFeeds argument**: It includes the list of feeds that are to be sequenced.

7. **FeedName suffixed by ‘CachedObjectsList’**:

Eg. ArgumentName: DivisionFeedCachedObjectsList

ArgumentValue: OrganizationDBCache,ShipNodeDBCache

This is the list of db classes that will be cache enabled/disabled(respectively on a EOF/SOF message) for a particular feed. As in the example, for a division feed, the tables that will be cache enabled/disabled will be YFS\_ORGANIZATION and YFS\_SHIP\_NODE. Each of the argument values given above will in itself be an argument where the value will be the DB class of that table. Eg: **Argument Name: OrganizationDBCache Argument Value: com.yantra.shared.dbclasses.YFS\_OrganizationDBCacheHome**. This is required as these values form the input to modifyCache API which actually does the enabling/disabling. If in future, to improve performance any new tables need to enabled/disabled in the cache, modify the ObjectsList arguments and then add/remove the entry in this list.

8. Catalog Index build required or not: If for a particular feed, the search index catalog needs to be rebuilt, there is a flag named CatalogIndexBuildReqd that needs to be set as “Y”. Default value is “N”.

**Note:** The SOF message format is <SOF FeedName=<name of feed name>. The EOF message format is <EOF YantraMessageGroupID=<name of feed name>.

