

```

/**
 * The <code>createCondensationHelper</code> method is used pack up the necessary information needed to create a condenssation function.
 * Then it issue a request to the QoS management to create this condensation function in the dataplane.
 * <BR>
 * @param placementSR Request that the condensation function will be placed on this SR.
 * @param pubName The name of the publisher, i.e. this condensation function for the variable produced by it.
 * @param variableName The name if the variable that this condensation function publishes.
 * @param outDataType The datatype of the variable that is published.
 * @param outUserType The user datatype (is a user defined type is used) of the variable that is published.
 * @param outTypeLen The lenght (bytes) of the published variable.
 * @param priority The priority of the published variable.
 * @param intervalLow The low publishing interval.
 * @param intervalHigh The high publishing interval.
 * @param inFilerHigh Should the input values be filtered for a max value.
 * @param inFilterHighVar If high filtering (input) then this is the max value.
 * @param inFilerLow Should the input values be filtered for a minimum value.
 * @param inFilterLowVar If low filtering (input) then this is the minimum value.
 * @param outFilerHigh Should the output value be filtered for a max value.
 * @param outFilterHighVar If high filtering (output) then this is the max value.
 * @param outFilerLow Should the output value be filtered for a minimum value.
 * @param outFilterLowVar If low filtering (output) then this is the minimum value.
 * @param triggerType The triggering type to be used.
 * @param triggerVar1 A variable that can be given to the triggering mechanism.
 * @param calculatorURI The location of the calculator class that should be used for this condensation function.
 * @param calculatorClassName The name of the calculator class that should be used for this condensation function.
 * @return Returns <code>CommConstants.COMMAND_OK</code> if the condensation function could be created, error code otherwise.
 */
public short createCondensation(String placementSR, String pubName, String variableName, short outDataType, int outUserType,
                               int outTypeLen, short priority, int intervalLow, int intervalHigh, boolean inFilerHigh,
                               int inFilterHighVar, boolean inFilerLow, int inFilterLowVar, boolean outFilerHigh, int outFilterHighVar,
                               boolean outFilerLow, int outFilterLowVar, short triggerType, int triggerVar1, String calculatorURI,
                               String calculatorClassName);

/**
 * The <code>addSubscription</code> method is to add a subscription as a input variable for the condensation function to be created.
 * <BR>
 * @param subscriberName The name of the subscriber.
 * @param pubName The name of the publisher.
 * @param variableName The name of the variable to be subscribed to.
 * @param modes The modes that this subscription will be valid in.
 * @param dataType The datatype of the variable that we subscribe to.
 * @param userType The type of the user registered type.
 * @param interval The interval that we wish to receive the events.
 * @param priority The priority of the subscription.
 * @param latency The latency that we would link for this subscription.
 * @param redundancy The redundancy that we would link for this subscription.
 * @return Returns <code>>true</code> if there is space to add this
 * subscription, <code>>false</code> otherwise.
 */
public boolean addSubscription(String subscriberName, String pubName, String variableName, int modes, short dataType, int userType,
                              int interval, short priority, short latency, short redundancy);

/**
 * The <code>removeCondensation</code> method is to unregister a previous created condensation function with the edgeStatusRouter.
 * <BR>
 * @param pubName The name of the publisher of the condensation variable.
 * @param variableName The name of the condensation variable.
 * @return Returns <code>CommConstants.COMMAND_OK</code> if the condensation function is removed, error code otherwise.
 */
public short removeCondensation(String pubName, String variableName);

/**
 * The <code>clearSubscriptions</code> method is used to clear all the subscriptions that have been adden in addSubscription.
 */
public void clearSubscriptions();
}

```