

```

package edu.wsu.gridstat.router.interfaces;

/**
 * <p>Title: CondFunInterface </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2004</p>
 * <p>Company: Washington State University </p>
 * @author Kjell Harald Gjermundrod
 * @version 1.0
 */

// Standard java packages.
import java.util.Vector;

// GridStat packages.
import edu.wsu.gridstat.router.common.EventHolder;
import edu.wsu.gridstat.router.condFunction.CondRoutingHolder;
import edu.wsu.gridstat.router.condFunction.CondStatusHolder;

public abstract interface CondFunInterface extends OutInterface
{
    /**
     * The <code>start</code> method is used to start this condensation function.
     * <BR>
     * @return Returns <code>>true</code> if we can load the calculator object and start the triggering thread, <code>>false</code> otherwise.
     */
    public boolean start();

    /**
     * The <code>stop</code> method is used to stop the thread.
     */
    public void stop();

    /**
     * The <code>addRoutingHolder</code> method is used to add a routing holder to this condensation function. This is a holder for one
     * of the subscription that is condensation function has.
     * <BR>
     * @param holder The routing holder that is added.
     */
    public void addRoutingHolder(CondRoutingHolder holder);

    /**
     * The <code>getRoutingHolders</code> method is used to get all the routing holders that this condensation function has.
     * <BR>
     * @return Returns a Vector of <code>CondRoutingHolder</code>.
     */
    public Vector getRoutingHolders();

    /**
     * The <code>doCalculation</code> method is used to perform the calculation of a new value and publish it as a status event.
     * <BR>
     * @return Returns <code>>true</code> if we could calculate a new value and publish it as a status event, <code>>false</code> otherwise.
     * @throws java.lang.InterruptedException
     */
    public boolean doCalculation() throws InterruptedException;

    /**
     * The <code>setValue</code> method is used to extract the value of an event that we have received.
     * <BR>
     * @param statusHolder The holder where we will set the value.
     * @param event The event that we will extract the value from.
     * @return Returns <code>>true</code> if the value is extracted and set, <code>>false</code> if the value is filtered.
     */
    public boolean setValue(CondStatusHolder statusHolder, EventHolder event);
}

```