

```

package edu.wsu.gridstat.publisher.interfaces;

/**
 * <p>Title: PublisherWithDV </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: Washington State University </p>
 * @author Kjell Harald Gjermundrod
 * @version 1.0
 */

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

// GridStat packages.
import edu.wsu.gridstat.publisher.common.PublisherException;

public interface PublisherWithDVInterface extends PublisherInterface
{
    /**
     * The <code>registerPublishInt</code> method is to register a publish with the edgeStatusRouter.
     * <BR>
     * @param variableName The name of the variable which event is to be published.
     * @param priority The priority of this status variable.
     * @param intervalLow The lowest interval that the message will be published.
     * @param intervalHigh The highest interval that the message will be published.
     * @param dvs A vector of derived values to be added to this publication.
     * @return Returns the variableId of the registered publication.
     * @throws PublisherException If something went wrong during the registration.
     */
    public int registerPublishInt(String variableName, short priority, int intervalLow,
                                int intervalHigh, Vector dvs) throws PublisherException;

    /**
     * The <code>registerPublishFloat</code> method is to register a publish with the edgeStatusRouter.
     * <BR>
     * @param variableName The name of the variable which event is to be published.
     * @param priority The priority of this status variable.
     * @param intervalLow The lowest interval that the message will be published.
     * @param intervalHigh The highest interval that the message will be published.
     * @param dvs A vector of derived values to be added to this publication.
     * @return Returns the variableId of the registered publication.
     * @throws PublisherException If something went wrong during the registration.
     */
    public int registerPublishFloat(String variableName, short priority, int intervalLow,
                                    int intervalHigh, Vector dvs) throws PublisherException;

    /**
     * The <code>registerPublishBoolean</code> method is to register a publish with the edgeStatusRouter.
     * <BR>
     * @param variableName The name of the variable which event is to be published.
     * @param priority The priority of this status variable.
     * @param intervalLow The lowest interval that the message will be published.
     * @param intervalHigh The highest interval that the message will be published.
     * @param dvs A vector of derived values to be added to this publication.
     * @return Returns the variableId of the registered publication.
     * @throws PublisherException If something went wrong during the registration.
     */
    public int registerPublishBoolean(String variableName, short priority, int intervalLow,
                                      int intervalHigh, Vector dvs) throws PublisherException;

    /**
     * The <code>registerPublishUserDefined</code> method is to register a publish with the edgeStatusRouter.
     * <BR>
     * @param variableName The name of the variable which event is to be published.
     * @param priority The priority of this status variable.
     * @param intervalLow The lowest interval that the message will be published.
     * @param intervalHigh The highest interval that the message will be published.
     * @param userType The type of the user registered type.
     * @param length The length in bytes of the user defined type.
     * @param dvs A vector of derived values to be added to this publication.
     * @return Returns the variableId of the registered publication.
     * @throws PublisherException If something went wrong during the registration.
     */
    public int registerPublishUserDefined(String variableName, short priority, int intervalLow,
                                          int intervalHigh, int userType, int length, Vector dvs) throws PublisherException;
}

```