

CommonsLogLogChute

This is a replacement of the [CommonsLogLogSystem](#) that is part of [VelocityTools](#) 1.1 and 1.2. It will be added to the trunk of [VelocityTools](#) after version 1.2 is released. This is dependent on Velocity 1.5 and provides additional logging facilities to match improvements in that version.

```
/*
 * Copyright 2005 The Apache Software Foundation.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.apache.velocity.tools.generic.log;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.velocity.runtime.RuntimeServices;
import org.apache.velocity.runtime.log.LogChute;

/**
 * Redirects Velocity's LogChute messages to commons-logging.
 *
 * <p>To use, first set up commons-logging, then tell Velocity to use
 * this class for logging by adding the following to your velocity.properties:
 *
 * <code>
 * runtime.log.logsystem.class = org.apache.velocity.tools.generic.log.CommonsLogLogChute
 * </code>
 * </p>
 *
 * <p>You may also set this property to specify what log/name Velocity's
 * messages should be logged to (example below is default).
 * <code>
 * runtime.log.logsystem.commons.logging.name = org.apache.velocity
 * </code>
 * </p>
 *
 * @author <a href="mailto:nathan@esha.com">Nathan Bubna</a>
 * @since VelocityTools 1.3
 * @version $Id: CommonsLogLogChute.java 71982 2004-02-18 20:11:07Z nbubna $
 */
public class CommonsLogLogChute implements LogChute
{

    /** Property key for specifying the name for the log instance */
    public static final String LOGCHUTE_COMMONS_LOG_NAME =
        "runtime.log.logsystem.commons.logging.name";

    /** Default name for the commons-logging instance */
    public static final String DEFAULT_LOG_NAME = "org.apache.velocity";

    /** the commons-logging Log instance */
    protected Log log;

    /** LogChute methods */

    public void init(RuntimeServices rs) throws Exception
    {
        String name =
```

```

        (String)rs.getProperty(LOGCHUTE_COMMONS_LOG_NAME);

    if (name == null)
    {
        name = DEFAULT_LOG_NAME;
    }
    log = LogFactory.getLog(name);
    log(LogChute.DEBUG_ID, "CommonsLogLogChute name is '" + name + "'");
}

/**
 * Send a log message from Velocity.
 */
public void log(int level, String message)
{
    switch (level)
    {
        case LogChute.WARN_ID:
            log.warn(message);
            break;
        case LogChute.INFO_ID:
            log.info(message);
            break;
        case LogChute.DEBUG_ID:
            log.debug(message);
            break;
        case LogChute.TRACE_ID:
            log.trace(message);
            break;
        case LogChute.ERROR_ID:
            log.error(message);
            break;
        default:
            log.debug(message);
            break;
    }
}

/**
 * Send a log message from Velocity with an error.
 */
public void log(int level, String message, Throwable t)
{
    switch (level)
    {
        case LogChute.WARN_ID:
            log.warn(message, t);
            break;
        case LogChute.INFO_ID:
            log.info(message, t);
            break;
        case LogChute.DEBUG_ID:
            log.debug(message, t);
            break;
        case LogChute.TRACE_ID:
            log.trace(message, t);
            break;
        case LogChute.ERROR_ID:
            log.error(message, t);
            break;
        default:
            log.debug(message, t);
            break;
    }
}

/**
 * Checks whether the specified log level is enabled.
 */
public boolean isLevelEnabled(int level)
{

```

```
switch (level)
{
    case LogChute.DEBUG_ID:
        return log.isDebugEnabled();
    case LogChute.INFO_ID:
        return log.isInfoEnabled();
    case LogChute.TRACE_ID:
        return log.isTraceEnabled();
    case LogChute.WARN_ID:
        return log.isWarnEnabled();
    case LogChute.ERROR_ID:
        return log.isErrorEnabled();
    default:
        return true;
}
}
```