

System Log



NextAge
CONSULTING

NextAge Consulting

Pete Halsted

110 East Center St. #1035

Madison, SD 57042

pete@thenextage.com

www.thenextage.com

www.thenextage.com/wordpress

Table of Contents

[Table of Contents](#)

[BSD 3 License](#)

[NextAge Consulting - Pete Halsted](#)

[General Information](#)

[Writing a Log Entry](#)

[Examples](#)

[Database Table](#)

[SystemLog](#)

[Project Code](#)

[Properties](#)

[UserIdVariable](#)

[MessageType](#)

[ProcedureName](#)

[ErrorDescription](#)

[SQLStatement](#)

[AdditionalInformation](#)

[Methods](#)

[WriteLog](#)

[Code Bricks](#)

[SysLog - System Log Record a Log Entry](#)

[Special Properties](#)

[SystemLogFile](#)

[UserIDField](#)

[MessageTypeField](#)

[ProcedureNameField](#)

[TimeStampField](#)

[ErrorDescriptionField](#)

[FileErrorDescriptionField](#)

[SQLStatementField](#)

[AdditionalInformation](#)

[Change Log](#)

[1.0 - Not Released](#)

BSD 3 License

Copyright (c) 2012, NextAge Consulting (www.thenextage.com)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the NextAge nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NextAge Consulting - Pete Halsted

Pete Halsted has been developing custom business management applications for small to medium-sized companies, since 1987. His focus is on client/server, distributed and cloud based development utilizing WinDev, WebDev, and PostgreSQL. Pete is a Clarion Certified Developer with 25 years in the industry, has spoken at several Developers conferences, and provided Developer training and mentoring on a one on one basis. He has served companies both large and small as Project Manager, Lead Architect, Lead Developer and Chief Technology Officer. Pete tours the country full-time by motor home with his wife and dog, enjoying the freedom provided by cloud based technologies. Pete is available for Project Management, Custom Design, Development, Training, and Speaking assignments. For more information please visit www.thenextage.com or follow his blog at www.thenextage.com/wordpress

General Information

The System Log Class was created to provide standard method of recording errors in a database table. Often we have users that call for support, but when asked what the error was, they say I don't know I just clicked enter. We use the System Log class to log all errors to a table, that way we can refer to the table for the time period in question and see what the actual errors were. We also use the function to record debug information in the same table based on a global variable. Many of the NextAge Open Source Classes also use this class.

Writing a Log Entry

Once the class is instantiated in your project code. Writing an entry is just a matter of setting whatever properties you wish to record and then calling the writelog method. There is a code template to assist with this (SysLog).

Examples

```
SysLog.MessageType = "Warning"  
SysLog.ProcedureName = MyWindow..Name  
SysLog.ErrorDescription = "There was a problem with an SQL Statement"  
SysLog.SQLStatement = sQuery  
SysLog.AdditionalInformation = "Yet More Information"  
SysLog.WriteLog()
```

Database Table

The class has been written in a generic fashion that allows the file name and field names to be set via properties if desired. Infact many of the fields themselves are optional and if the property is set to an empty string then the field will be ignored completely. The class also uses Hxxxx commands for all file access, meaning that the class should work with any database that WX supports. The suggested definition for this file follows. If you do not use this definition you will have to use the associated properties to point to the correct values (*See the Special Properties*).

SystemLog

- SystemLogID
 - Automatic ID
- UserID
 - Is populated with the current value of the variable set via the UserIdVariable Property
 - Integer
- MessageType
 - Text (10)
- ProcedureName
 - Text (50)
- TimeStamp
 - Is automatically populated with the Current Date and Time
 - DateTime
- ErrorDescription
 - Text (5000)
- FileErrorDescription
 - Is automatically populated with the HErrorInfo if there is any.
 - Text (5000)
- SQLStatement
 - Text (5000)
- AdditionalInformation
 - XML (Text Memo)

Project Code

The class should be initialized in your project initialization code. If you are overriding the file or fieldnames it should be done at this time.

1. `SysLog` is `SystemLogClass`
 - Instantiates an instance of the class global for the entire project.
2. `SysLog.UserIdVariable = "Glo.UserSysId"`
 - Sets the name of the variable that contains the current users unique identifier.

Properties

Note: Private Properties are not documented.

AdditionalInformation

A text memo field, we use this to store XML data when we want to store structured debug information.

ErrorDescription

The main text of the error.

MessageType

For example, Warning, Stop, Debug, Performance, Security, PCI, etc.

ProcedureName

Very useful for debugging. For windows etc. remember you can use mywindow..name

SQLStatement

When using SQL statements in the project, it is very helpful when debugging to have the actual SQL statement that was generated, so we recorded it in a separate field.

UserIdVariable

Variable Name that contains the currently logged in users unique id. If not set then UserID will not be recorded in the log.

Methods

Note: Private Methods are not documented.

WriteLog

Writes a log entry to the database table.

Code Bricks

SysLog - System Log Record a Log Entry

Code stub for writing a log entry.

Special Properties

The below properties can be used to change the default file name and field names used.

SystemLogFile

The name of the System Log file. Default = "SystemLog"

UserIDField

Default = "UserID"

MessageTypeField

Default = "MessageType"

ProcedureNameField

Default = "ProcedureName"

TimeStampField

Default = "TimeStamp"

ErrorDescriptionField

Default = "ErrorDescription"

FileErrorDescriptionField

Default = "FileErrorDescription"

SQLStatementField

Default = "SQLStatement"

AdditionalInformation

Default = "AdditionalInformation"

Change Log

1.0 - January 16, 2013

Initial Release