WRIA-1 DSS

TECHNICAL DOCUMENTATION MANUAL

Table of Contents

Installation Technology Overview....1
Detailed Explanation of Installations and Merge Modules....5
Technical Documentation: The 1D Habitat Viewer....21
Technical Documentation: The 2D Habitat Viewer....37
Technical Documentation: Best Management Practices Tool....50
Technical Documentation: the Nooksack DBMS/LaunchPad....56
Technical Documentation: The Habitat Time Series Model....67
Technical Documentation: Lake Whatcom Water Quality Model....85
Technical Documentation: Land Cover Changer Tool....90
Technical Documentation: Land Cover Summarizer Tool....93
Technical Documentation: The Macroinvertebrate Data Viewer....96
Technical Documentation: The Photo Viewer....131
Technical Documentation: The Periodicity Viewer/Editor....140
Technical Documentation: Water Quantity Model....154
Technical Documentation: The Temperature and DO Flags Model....167
Technical Documentation: The Time Series Analyst....178
Technical Documentation: Watershed Characterization Report Generator....189
Technical Documentation: The Well Log Data Viewer....236
Technical Documentation: Course Resolution Water Quality Model....245

Installation Technical Documentation for the WRIA-1 DSS

created by Christopher Michaelis on October 19, 2004

Installation Technology Overview

Software these days cannot simply be copied to a destination computer and run. Modern software components are complex and intricate, often having a long list of dependencies which must be properly installed and registered with windows. In an effort to keep track of software and dependencies, we have organized the components of the WRIA-1 DSS using pre-built installation modules (merge modules) and InstallShield installations. There are many software packages available to help in organizing software to be installed; we have settled on InstallShield Express 5.1 as an excellent balance of cost and features. We utilize Microsoft Visual Studio to build merge modules, which is the only significant lack of functionality in InstallShield Express.

The software we use for installations is explained briefly below.

InstallShield Express 5.1

InstallShield Express is a product intended specifically for developing installations. These installations can run on any version of Windows (InstallShield Express 5.1 does not support Linux or PalmOS or Windows CE).

The installations produced by InstallShield express are capable of installing files to a system, installing merge modules (see below for a definition of merge modules), creating shortcuts to files, adding windows registry entries, and performing custom actions (DLL calls or executable files). InstallShield Express provides some flexibility in configuring the user interface which will be presented during an installation. The “Express” edition of InstallShield is a bit limited, but the price tag difference between this and the “Full” edition makes using it very worthwhile. The features that we’re missing by not using the “Full” edition are basically not having complete and utter control over the installation process (i.e., changing the order in which dialogs appear, creating completely new dialogs, or performing special functions during the middle of the installation). InstallShield Express is also not capable of producing merge modules; we use Microsoft’s Visual Studio to accomplish this.

InstallShield Express does still allow “Custom Actions” at the end of the installation; a custom action is a call to an external executable file or DLL. This file can be on the installation media or installed with the product.

Using InstallShield Express is a fairly straightforward process. On opening the project (the .ise file), you’ll see a panel on the left-hand side with options such as “Redistributables”, “Files”,

"Custom Actions", etc. Clicking on each of these in turn will display that option in the main window. They are each fairly self-explanatory, with the following cautions:

First, "redistributables" refers to merge modules. This is important to point out so that you realize that this is where the merge modules are selected. The merge modules that appear here are the modules which are present in the directory C:\Program Files\InstallShield\Express\Objects. Therefore, after putting together a merge module, in order for InstallShield to see and use it, you must place the file (*.msm) in that directory.

Secondly, when you build the installation (using the menu option or the toolbar icon), the installation will build only the last media type which you had selected. If you have configured more than one media type, they will NOT all be built – you must click on each one in turn, then click the build icon.

Thirdly, when you're trying to locate your built installation, it will usually be buried under many levels of subdirectories. Suppose you're working with installation X.ise; there will be a subdirectory entitled X next to X.ise. Under this directory, there will be an Express subdirectory. Under this directory, there will be a subdirectory for each of the media types that you have built. Suppose you built a "CDROM" installation; there will be a subdirectory entitled CDROM. In this subdirectory, there will be about three directories; the one you care about is entitled "Disk Images". In this directory, there will be a DISK1, DISK2, and so forth.

Fourthly, if you build an installation which spans multiple disks or CDs, you must label those CDs to match the subdirectory that the files were in. For example, when burning the files from DISK1 onto a CD, you must set that CD's volume label to DISK1. This applies to DISK2, DISK3, and so forth. If you do not set the volume label of the CD properly, then the installation will not recognize the CD as being the proper disk. The installation will ask for disk "2". Even if the CD you insert has the proper files on it, if the label is not DISK2 the installation will not accept the CD.

Fifthly, there are a few file restrictions that you need to be aware of. You should never place the following files in the C:\Program Files\MapWindow\Plugins subdirectory (or any subdirectory underneath Plugins) :

a. stdole.dll
b. MapWinInterfaces.dll
c. Microsoft.VisualBasic.Compatibility.dll
d. MapWinGIS.ocx

These items may appear in the same directory as MapWindow.exe. Files such as AxInterop.MapWinGIS.dll and Interop.MapWinGIS.dll can be in the plugins subdirectory, but not MapWinGIS.ocx itself.

With these tips, you should be able to use InstallShield express fairly easily. If you have questions on using InstallShield, the software comes with a 'Quick Start' guide which can be quite useful. There is also in-depth documentation available in the help menu.

Merge Modules

Often, it is desirable to build a reusable installation package for a given component. Take, for instance, MapWindow 3.1. This software package is used as the basis for a wide variety of applications and purposes. It doesn't make sense to put together a MapWindow installation every time it is needed. To solve this problem, we use Merge Modules.

A merge module is an installation module which is usually self-contained and doesn't need anything else to have the complete piece. For instance, the MapWindow merge module contains the MapWindow software and all of the dependencies that MapWindow needs.

Merge modules are also often used to group together a component and all of that component's dependencies. An example of this is the AddFlow flowcharting component. This component is a single file, but it has a few dependencies. So, this single file and all of its dependencies are packaged together into AddFlow.msm, which is a self-contained package providing the AddFlow charting component and everything it needs. Any application which needs this component, then, can include the merge module in its installation, without needing to worry if all of the dependencies and such have also been included. This is a great way to modularize components and simplify a large installation.

One more example of merge modules is the Time Series Analyst merge module. This software requires MapWindow, since it's a MapWindow plugin. However, the merge module does NOT include MapWindow. It does, however, include all of the other dependencies – Gigasoft Proessentials, Windows components, et cetera. This provides a bit more flexibility – if you want to put together an installation to give the Time Series Analyst to somebody who you already know has MapWindow, you could put together an InstallShield installation containing the Time Series merge module and a database with a few shapefiles, excluding MapWindow on purpose. If you wanted to give the installation to users without MapWindow, you could simply include the MapWindow merge module as well.

Using merge modules slows down development slightly in the beginning, as you must decide what needs to be put into separate merge modules and you must create "wrapper" installations to hold these merge modules. However, this small time investment in the beginning speeds up future installations considerably and pays for itself several times over.

Often, you may build a merge module for inclusion in large installations, but you also wish to be able to install only the component in the merge module without installing other things as well. The Time Series Analyst is another ideal example of this – sometimes, you may want to install only the Time Series Analyst. Since a merge module cannot be installed directly, you must create an InstallShield installation which installs nothing but this merge module. This is referred to as a "Wrapper" installation for that merge module, since its sole purpose is to wrap that merge module, making it installable.

Occasionally in this document when describing merge modules, you'll see a reference to a "wrapper" installation location. This is showing where a wrapper installation for merge module has already been created.

Additionally, a merge module may contain other merge modules. A good example of this is the Model Manager merge module, which also contains the AddFlow merge module. Suppose that you had another merge module, X.msm, which also contained the AddFlow merge module; if you build an InstallShield installation that includes both X.msm and the Model Manager merge module, InstallShield will automatically detect that AddFlow is included twice and will only use the files once. This is particularly important with very common components that are duplicated many times in an installation. This phase of building the installation is called “merging”, hence the term “merge module”.

Visual Studio 2002 (or 2003)

The majority of the software for the WRIA-1 DSS has been created using various versions of Microsoft’s Visual Studio; the scope of this section is not focused on software, but rather using Visual Studio for the purpose of building Merge Modules.

Visual Studio can be used to build complete application packages, but I find Visual Studio severely limited in terms of customizing the installation and making it professional looking. I use Visual Studio only for merge modules, and this is because InstallShield Express will not build merge modules. (InstallShield Developer edition will allow you to build merge modules; however, it is considerably more expensive than InstallShield Express, and Visual Studio, a tool we already have, can do Merge Modules perfectly well.)

All of the merge modules have been created with Visual Studio 2002, and can therefore be opened with Visual Studio 2002 or 2003. To create a new merge module from scratch, choose New Project from the File menu. Select “Setup or Deployment” project, then choose Merge Module. You’ll get an empty list; from here, you may add an executable (dependencies will automatically be determined and added). You may also add any other file type. Do this by either dragging a file in, or right-clicking and choosing Add File.

The path which files are configured to install to in the merge module will carry over to the installation that uses the merge module; the installer using the merge module cannot specify a different path to install the files, unless you place files in the “retargetable” folder. Therefore, place the files carefully, paying attention to directory structure.

You may change where the merge module output goes by right-clicking on the project in the right-hand pane. Choose Properties, and you’ll see a window with a text box labeled “Project output”. Change this to the place you’d like the merge module to go.

Build the merge module by choosing Build from the menu. When the file has been built, remember to place the resulting .msm in the proper location. For InstallShield Express, this is “c:\Program Files\InstallShield\Express\Objects”. If you’re using Visual Studio or any other installation utility, you’ll likely need to place the merge module in “c:\Program Files\Common Files\Merge Modules”. After rebuilding an existing merge module, simply overwrite the old one.

Detailed Explanation of Installations and Merge Modules

Here, all of the installations and merge modules that are used within the WRIA-1 DSS installation package will be listed and explained. For those components which were built at USU, the contents of the merge module and the purpose of each file will be listed. Components which were not built at USU (i.e., Micosoft component merge modules) will list only what that component is. Be aware that some of the files in these merge modules may change slightly over time, but the overall contents and purpose will remain the same.

WRIA-1 Final Deliverable Standard Edition Installshield Installation

(installation\WRIA-1 Final Deliverable Standard\WRIA-1 Final Deliverable.ise)

This is the main "container" for all components, data, and merge modules which need to be installed with the Nooksack DSS. This InstallShield installation packages everything together and generates the set of install CDs to be distributed to end users.

This is the Standard edition of the installation, which means that it includes the following components and objects:

<u>Merge Modules</u> *(explanation of these component will follow in this document)*

- AddFlow Flowchart Components (AddFlow.msm)
- DBMS Standard Edition (DBMSStandard.msm)
- Gigasoft Proessentials Charting Components (Gigasoft.msm)
- MapWindow 3.1 (MapWindow3.1.msm)
- Model Manager (Model Manager.msm)
- Model Manager Elements (ModelManagerElements.msm)
- Macroinvertibrate Data Viewer (MIVViewer.msm)
- Physical Habitat Simulation 1D Viewer (PHabSim1DViewer.msm)
- Physical Habitat Simulation 2D Viewer (PS2DViewer.msm)
- Photo Viewer (mwPhotoViewer.msm)
- Time Series Data Analyst (mwTimeSeries.msm)
- Watershed Characterization Report (mwWatershedChar.msm)
- Well Log Data Viewer (mwWellviewer.msm)
- Crystal Reports 8.5 Runtime Components (CrystalReports85.msm)
- Crystal Reports .NET Data Access (Crystal_Database_Access2003.msm)
- Crystal Reports .NET English (Crystal_Database_Access2003_enu.msm)
- Crystal Reports .NET Managed Code (Crystal_Managed2003.msm)
- Seagate Crystal Reports Keycode Manager (Crystal_regwiz2003.msm)
- Data Access Objects 3.60 (DAO360.msm)
- Distributed Component Object Model 95 (DCOM95.msm)
- Microsoft Foundations Classes 6.0 Libraries (MFC42.msm)
- Microsoft C Runtime Library 6.0 (MSVCRT.msm)
- Microsoft C++ Runtime Library 6.0 (MSVCP60.msm)
- Microsoft Chart Control 6.0 (MSCHRT20.MSM)

Microsoft Chart VB Control (VB_Control_mschart_RTL_X86_---.msm)
Microsoft Common Dialog Control 6.0 (COMDLG32.msm)
Microsoft Component Category Manager Library (COMCAT.msm)
Microsoft Data Access Components (MDAC) 2.5 (MDAC25.msm)
MDAC 2.6 (MDAC26.msm)
MDAC 2.7 (MDAC27enu.msm)
Microsoft FlexGrid Control 6.0 (MSFLXGRD.msm)
Microsoft OLE 2.40 for 95/NT4.0 (OLEAUT32.msm)
Microsoft Typelib Information Library (tlbinf32.msm)
Microsoft Windows Common Controls 6.0 (MSCOMCTL.msm)
Microsoft Windows Common Controls-2 6.0 (MSCOMCT2.msm)
OLE Database Access 2.1 (OLEDB21.msm)
SQL Distributed Management Object (SQL-DMO.msm)

Shortcuts
Programs Menu \ MapWindow
MapWindow -> [MAPWINDOW]MapWindow.exe
User's Guide -> [HELP]MapWindow31.chm
Sample Project -> [United States]UnitedStates.mwprj
Programs Menu \ WRIA-1 DSS
Launch the DSS -> [LAUNCHPAD]DBMS.exe
User Documenation -> [USER_DOCUMENTATION]
Desktop
WRIA-1 DSS LaunchPad -> [LAUNCHPAD]DBMS.exe

Windows Registry
HKEY_LOCAL_MACHINE \ Software \ Microsoft \ Windows \ CurrentVersion
\ Run \ "Start SQL Server" (String Value)
= "scm –action 1 –service MSSQLServer –silent 1"
\RunOnce \ "AttachAllDatabases" (String Value)
= "C:\Program Files\Microsoft SQL Server\MSSQL\Data\AttachAll.bat

Dialogs Included in Installation
Install Welcome
Setup Progress
Setup Complete Success

Custom Actions
ScheduleReboot (After Setup Complete Success Dialog)
Source Location: Built-In Library Function

MSDEInstaller (After Setup Complete Success Dialog)
Source Location: Installed With Product
File Name: [TempFolder]MSDE\setup.exe
Command Line: (on next line)
INSTANCENAME=MSSQLSERVER BLANKSAPWD=1 SECURITYMODE=SQL

Files Installed (Always Install feature)

[ProgramFilesFolder]

\LaunchPad

Nothing installed here, but the path must exist for shortcuts.
The files installed here are put here by the DBMS merge module.

\MapWindow\Help

Again, nothing installed – path is here for shortcuts. The
Merge module installs the needed files.

\MapWindow\Sample Data\UnitedStates

Again, this is for the shortcuts. The MapWindow merge module
Places the sample data here.

\Microsoft SQL Server\MSSQL\Data

These are the databases and the database attachers.

\Microsoft SQL Server\MSSQL\Data\Resources\1033\SQLDMO.rll

This is the English language resource file for SQL Server.

\WRIA-1_DSS*.*

This is the DSS Data and User Documentation.

[TempFolder]

\MSDE*.* *This is the MSDE installation program*

Special Actions

If this installation is going to be used on a computer with multiple users, you'll need to perform the following steps:

1. Build the installation normally.
2. Download and install ORCA, a tool for editing installation databases. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/orca_exe.asp
3. Open ORCA. Open the .msi file associated with the installation that you just built. This will be on the first disk of your installation. This will probably be called "WRIA-1 Final Deliverable Standard Edition.msi".
4. Click on the Components table; you'll see a long list of data.
5. Find the component called "Global_Controls_MSCOMCT2OCX…". This component is associated with GUID "3207D1B8-80E5-11D2-B95D-006097C4DE24".
6. Find the "keypath" field for this component. Click into that field, and clear out its contents. (Clear out only the KEYPATH portion; don't delete the entire row!)
7. Close and save the .msi file.
8. The installation package may now be safely distributed to end users.

WRIA-1 Final Deliverable Administrative Edition Installshield Installation

(installation\WRIA-1 Final Deliverable Administrative\WRIA-1 Final Deliverable.ise)

This is the main "container" for all components, data, and merge modules which need to be installed with the Nooksack DSS. This InstallShield installation packages everything together and generates the set of install CDs to be distributed to end users.

This is the Administrative edition of the installation, which means that it includes the following components and objects:

Merge Modules (explanation of these component will follow in this document)
AddFlow Flowchart Components (AddFlow.msm)
DBMS Administrative Edition (DBMSAdmin.msm)
Gigasoft Proessentials Charting Components (Gigasoft.msm)
MapWindow 3.1 (MapWindow3.1.msm)
Model Manager (Model Manager.msm)
Model Manager Elements (ModelManagerElements.msm)
Macroinvertibrate Data Viewer (MIVViewer.msm)
Physical Habitat Simulation 1D Viewer (PHabSim1DViewer.msm)
Physical Habitat Simulation 2D Viewer (PS2DViewer.msm)
Photo Viewer (mwPhotoViewer.msm)
Time Series Data Analyst (mwTimeSeries.msm)
Watershed Characterization Report (mwWatershedChar.msm)
Well Log Data Viewer (mwWellviewer.msm)
Crystal Reports 8.5 Runtime Components (CrystalReports85.msm)
Crystal Reports .NET Data Access (Crystal_Database_Access2003.msm)
Crystal Reports .NET English (Crystal_Database_Access2003_enu.msm)
Crystal Reports .NET Managed Code (Crystal_Managed2003.msm)
Seagate Crystal Reports Keycode Manager (Crystal_regwiz2003.msm)
Data Access Objects 3.60 (DAO360.msm)
Distributed Component Object Model 95 (DCOM95.msm)
Microsoft Foundations Classes 6.0 Libraries (MFC42.msm)
Microsoft C Runtime Library 6.0 (MSVCRT.msm)
Microsoft C++ Runtime Library 6.0 (MSVCP60.msm)
Microsoft Chart Control 6.0 (MSCHRT20.MSM)
Microsoft Chart VB Control (VB_Control_mschart_RTL_X86_---.msm)
Microsoft Common Dialog Control 6.0 (COMDLG32.msm)
Microsoft Component Category Manager Library (COMCAT.msm)
Microsoft Data Access Components (MDAC) 2.5 (MDAC25.msm)
MDAC 2.6 (MDAC26.msm)
MDAC 2.7 (MDAC27enu.msm)
Microsoft FlexGrid Control 6.0 (MSFLXGRD.msm)
Microsoft OLE 2.40 for 95/NT4.0 (OLEAUT32.msm)
Microsoft Typelib Information Library (tlbinf32.msm)
Microsoft Windows Common Controls 6.0 (MSCOMCTL.msm)

Microsoft Windows Common Controls-2 6.0 (MSCOMCT2.msm)
OLE Database Access 2.1 (OLEDB21.msm)
SQL Distributed Management Object (SQL-DMO.msm)

Shortcuts
Programs Menu \ MapWindow
MapWindow -> [MAPWINDOW]MapWindow.exe
User's Guide -> [HELP]MapWindow31.chm
Sample Project -> [United States]UnitedStates.mwprj
Programs Menu \ WRIA-1 DSS
Launch the DSS -> [LAUNCHPAD]DBMS.exe
User Documenation -> [USER_DOCUMENTATION]
Technical Documentation -> [INSTALLDIR]Technical Documentation
Source Code -> [WindowsVolume]Dev
Desktop
WRIA-1 DSS LaunchPad -> [LAUNCHPAD]DBMS.exe

Windows Registry
HKEY_LOCAL_MACHINE \ Software \ Microsoft \ Windows \ CurrentVersion
\ Run \ "Start SQL Server" (String Value)
= "scm –action 1 –service MSSQLServer –silent 1"
\RunOnce \ "AttachAllDatabases" (String Value)
= "C:\Program Files\Microsoft SQL Server\MSSQL\Data\AttachAll.bat

Dialogs Included in Installation
Install Welcome
Setup Type
Setup Progress
Setup Complete Success

Custom Actions
ScheduleReboot (After Setup Complete Success Dialog)
Source Location: Built-In Library Function

MSDEInstaller (After Setup Complete Success Dialog)
Source Location: Installed With Product
File Name: [TempFolder]MSDE\setup.exe
Command Line: (on next line)
INSTANCENAME=MSSQLSERVER BLANKSAPWD=1 SECURITYMODE=SQL

Files Installed (Always Install feature)
[ProgramFilesFolder]
\LaunchPad
Nothing installed here, but the path must exist for shortcuts.
The files installed here are put here by the DBMS merge module.
\MapWindow\Help

Again, nothing installed – path is here for shortcuts. The Merge module installs the needed files.

\MapWindow\Sample Data\UnitedStates

Again, this is for the shortcuts. The MapWindow merge module Places the sample data here.

\Microsoft SQL Server\MSSQL\Data

These are the databases and the database attachers.

\Microsoft SQL Server\MSSQL\Data\Resources\1033\SQLDMO.rll

This is the English language resource file for SQL Server.

\WRIA-1_DSS*.*

This is the DSS Data and User Documentation.

[TempFolder]

\MSDE*.* *This is the MSDE installation program.*

Files Installed (SourceAndTechDocs feature)

[WRIA-1_DSS]

\Technical Documentation*.*

This is the technical documentation for all reviewable products, including this document. (May not be present in Alpha releases)

[WindowsVolume]

\Dev*.*

This is the source code for all reviewable products. (May not be present in Alpha releases)

Special Actions

If this installation is going to be used on a computer with multiple users, you'll need to perform the following steps:

1. Build the installation normally.
2. Download and install ORCA, a tool for editing installation databases.
3. Open ORCA. Open the .msi file associated with the installation that you just built. This will be on the first disk of your installation. This will probably be called "WRIA-1 Final Deliverable Administrative Edition.msi".
4. Click on the Components table; you'll see a long list of data.
5. Find the component called "Global_Controls_MSCOMCT2OCX…". This component is associated with GUID "3207D1B8-80E5-11D2-B95D-006097C4DE24".
6. Find the "keypath" field for this component. Click into that field, and clear out its contents. (Clear out only the KEYPATH portion; don't delete the entire row!)
7. Close and save the .msi file.
8. The installation package may now be safely distributed to end users.

AddFlow Flowchart Components (AddFlow.msm)

(installation\AddFlow\Merge Module\AddFlow\AddFlow.sln)
This is a merge module built with Visual Studio to provide the AddFlow™ components which are used by the Model Manager. All of the files in this merge module are installed to the Windows System directory (usually c:\windows\system32).

Files Included

ADDFLOW3.LIC *The file storing the USU AddFlow component license.*
AddFlow3.ocx *The actual ActiveX AddFlow component.*
Mfc42.msm *Microsoft Foundations Classes 4.2, an AddFlow dependency.*
Msvcrt.msm *Microsoft Visual C Runtime Library, an AddFlow dependency.*
Oleaut32.msm *OLE Automation libraries, an AddFlow dependency.*

DBMS Standard Edition (DBMSStandard.msm)

(installation\DBMS\Merge Module\DBMSStandard\DBMSStandard.sln)
(InstallShield Wrapper at installation\DBMS\DBMSStandard.ise)

This is a merge module built with Visual Studio to provide the standard edition of the DBMS/LaunchPad. This is the Standard edition, meaning it does not allow remote data editing or LaunchPad reconfiguration. All of the files in this merge module are installed to C:\Program Files\LaunchPad.

Files Included

Adodb.dll *Used for database access (Active Data Object)*
axinterop.mscomctl2.dll *.NET wrapper for ActiveX component mscomctl2*
axinterop.msflexgridlib.dll *.NET wrapper for ActiveX component MSFlexGrid*
crystal_database_Access_2003.msm *Crystal Reports Database Libraries*
CrystalDecisions.CrystalReports.Engine.dll *Crystal Reports Core Engine*
CrystalDecisions.ReportSource.dll *Crystal Reports Data Source Locator*
CrystalDecisions.Shared.dll *Shared Libraries for Crystal Reports*
CrystalDecisions.Windows.Forms.dll *Form designer for Crystal Reports*
CrystalKeyCodeLib.dll *Registration Library for Crystal Reports*
Dotnetfxredist_x86_enu.msm *Suppressed (not installed - .NET Framework)*
Dtspkg.dll *Used for database management (DTS = Distributed Transactions)*
Interop.DTS.dll *.NET wrapper for Dtspkg.dll*
Interop.MSComCtl2.dll *.NET Wrapper for MSComCtl2.dll*
Interop.MSFlexGridLib.dll *.NET Wrapper for MSFlexGrid*
Interop.SQLDMO.dll *.NET Wrapper for SQLDMO component*
LifestagePlotter.dll *.NET Component for editing Fish Lifestage data*
Mscomct2.msm *Microsoft Common Controls library; used for animation controls*
Msflxgrd.msm *Microsoft FlexGrid control; used for data editing interface*
SQLDMO.dll *SQL Distribted Management Object – used to manage SQL Server*

Stdole.dll *Standard OLE interface library*
DBMS.conf *DBMS Configuration File (prebuilt for Nooksack)*
DBMS.exe *DBMS Executable itself – Standard Edition*
DBMS.pdb *DBMS Program Debug Library – helpful for debugging purposes*
ICSharpCode.SharpZipLib.dll *Compression library – used to speed up network*
MDAC27ENU.msm *Data Access components for ODBC data access*

DBMS Administrative Edition (DBMSAdmin.msm)

(installation\DBMS\Merge Module\DBMSAdmin\DBMSAdmin.sln)
(InstallShield Wrapper at installation\DBMS\DBMSAdmin.ise)

This is a merge module built with Visual Studio to provide the administrative edition of the DBMS/LaunchPad. This is the Administrative edition, meaning it allows remote data editing and LaunchPad reconfiguration. All of the files in this merge module are installed to C:\Program Files\LaunchPad.

Files Included
Adodb.dll *Used for database access (Active Data Object)*
axinterop.mscomctl2.dll *.NET wrapper for ActiveX component mscomctl2*
axinterop.msflexgridlib.dll *.NET wrapper for ActiveX component MSFlexGrid*
crystal_database_Access_2003.msm *Crystal Reports Database Libraries*
CrystalDecisions.CrystalReports.Engine.dll *Crystal Reports Core Engine*
CrystalDecisions.ReportSource.dll *Crystal Reports Data Source Locator*
CrystalDecisions.Shared.dll *Shared Libraries for Crystal Reports*
CrystalDecisions.Windows.Forms.dll *Form designer for Crystal Reports*
CrystalKeyCodeLib.dll *Registration Library for Crystal Reports*
Dotnetfxredist_x86_enu.msm *Suppressed (not installed - .NET Framework)*
Dtspkg.dll *Used for database management (DTS = Distributed Transactions)*
Interop.DTS.dll *.NET wrapper for Dtspkg.dll*
Interop.MSComCtl2.dll *.NET Wrapper for MSComCtl2.dll*
Interop.MSFlexGridLib.dll *.NET Wrapper for MSFlexGrid*
Interop.SQLDMO.dll *.NET Wrapper for SQLDMO component*
LifestagePlotter.dll *.NET Component for editing Fish Lifestage data*
Mscomct2.msm *Microsoft Common Controls library; used for animation controls*
Msflxgrd.msm *Microsoft FlexGrid control; used for data editing interface*
SQLDMO.dll *SQL Distribted Management Object – used to manage SQL Server*
Stdole.dll *Standard OLE interface library*
DBMS.conf *DBMS Configuration File (prebuilt for Nooksack)*
DBMS.exe *DBMS Executable itself – Administrative Edition*
DBMS.pdb *DBMS Program Debug Library – helpful for debugging purposes*
ICSharpCode.SharpZipLib.dll *Compression library – used to speed up network*
MDAC27ENU.msm *Data Access components for ODBC data access*

Gigasoft Proessentials Charting Components (Gigasoft.msm)

(Installation\GigaSoft\Merge Module\Gigasoft\Gigasoft.sln)

This is a merge module used to provide all of the Gigasoft ProEssentials graphing components, both versions 3 and 4, and all of their dependencies. This was built in response to consistently missing dependencies for these controls. All of these files are placed in the Windows System32 directory; some are registered, some are not.

Files Included

Pe3do16a.ocx *Contains a ProEssentials ActiveX control.*
Pe3do32a.ocx *Contains a ProEssentials ActiveX control.*
Pego16a.ocx *Contains a ProEssentials ActiveX control.*
Pego32a.ocx *Contains a ProEssentials ActiveX control.*
Pego32b.ocx *Contains a ProEssentials ActiveX control.*
Pegrp16a.dll *Contains requisite libraries used by the ActiveX controls.*
Pegrp32a.dll *Contains requisite libraries used by the ActiveX controls.*
Pegrp32b.dll *Contains requisite libraries used by the ActiveX controls.*
Pepco16a.ocx *Contains a ProEssentials ActiveX control.*
Pepco32a.ocx *Contains a ProEssentials ActiveX control.*
Pepso16a.ocx *Contains a ProEssentials ActiveX control.*
Pepso32a.ocx *Contains a ProEssentials ActiveX control.*
Pesgo16a.ocx *Contains a ProEssentials ActiveX control.*
Pesgo32a.ocx *Contains a ProEssentials ActiveX control.*
Pesgo32b.ocx *Contains a ProEssentials ActiveX control.*

MapWindow 3.1 (MapWindow3.1.msm)

(Installation\MapWindow31\Merge Module\MapWindow3.1\MapWindow3.1\MapWindow3.1.sln)

This is a merge module intended to wrap MapWindow 3.1 and all of its immediate dependencies together. All of the files are placed in C:\Program Files\MapWindow.

There are a bunch of files ending in .bgd, .shp, .shx, and .dbf that are not listed here. These are data items used by the sample project (USA).

Files Included

Interop.MapWinGIS.dll *.NET wrapper for MapWinGIS ActiveX control.*
AxInterop.MapWinGIS.dll *.NET wrapper for MapWinGIS ActiveX control.*
Comdlg32.msm *Common Dialog controls (Save As, Open…)*
Default.mwcfg *The default MapWindow configuration file.*
DevComponents.DotNetBar.dll *Component used for MapWindow menus.*
GridWizard.dll *MapWindow Grid Wizard plugin for processing grids.*
MapWindow.exe *MapWindow main executable program.*

MapWindow31.chm *Help file for MapWindow.*
MapWinGIS.ocx *MapWindow core mapping ActiveX component.*
MapWinInterfaces.dll *MapWindow plugin interface definitions.*
Microsoft.VisualBasic.Compatibility.dll *.NET library for handling VB6 plugins*
Mscomctl.msm *Microsoft Common Controls library*
MSIMG32.dll *Microsoft Image library*
mwIdentifier.dll *MapWindow Feature Identifier plugin for examining maps*
ShapefileEditor.dll *MapWindow Shapefile Editor plugin*
TableEditor.mw.dll *MapWindow Shapefile Table Editor plugin.*
Tlbinf32.msm *Microsoft Type Library Information object*
UnitedStates.mwprj *Example Project for USA (Also includes many data items)*

Model Manager (Model Manager.msm)

(Installation\Model Manager\Merge Module\Model Manager\Model Manager.sln)

This merge module includes the Model Manager MapWindow plugin, and it also includes the AddFlow components, which it uses. The files are placed in C:\Program Files\MapWindow\Plugins\ModelManager.

Files Included
AxInterop.Addflow3Lib.dll *.NET wrapper for Addflow ActiveX control.*
DevComponents.DotNetBar.dll *Used for the dockable menu bars and model list*
Dotnetfxredist_x86_enu.msm *Suppressed (not installed)*
Interop.AddFlow3Lib.DLL *.NET library wrapper for AddFlow*
Interop.DSSIntfcLib.dll *.Net Library wrapper for the DSS Model Interface*
Stdole.dll *Suppressed (not installed)*
System.dll *Suppressed (not installed)*
System.drawing.dll *Suppressed (not installed)*
System.xml.dll *Suppressed (not installed)*
AddFlow.msm *The AddFlow component used to create the scenario layout*
ModelManager.dll *The Model Manager MapWindow plugin*
ModelManager.pdb *Program Debug Database for the Model Manager*
mwDSS.xmlcf *Configuration file for the Nooksack DSS*
ScenarioBuilder.xml *Configuration file for the Scenario Builder layout*

Model Manager Elements (ModelManagerElements.msm)

(Installation\Model Manager Elements\Merge Module\ModelManagerElements\ModelManagerElements.sln)

This merge module contains all of the model elements (filters, data editors, and models) that the Model Manager can use. They're all lumped together in this merge module because it doesn't

make sense to separate them all into separate installations. The files are placed in C:\Program Files\MapWindow\Plugins\ModelManager\Elements.

Files Included

Interop.DSSIntfcLib.dll *.NET wrapper for the DSS Model Interface library.*
Interop.MapWinGIS.dll *.NET wrapper for the MapWindow Mapping component.*
DiversionChanger.dll *The diversion changer model element*
DiversionChanger.pdb *Program debug database for the above*
Interop.MapWinGIS.dll *.NET wrapper for the MapWindow Mapping component.*
LandCoverSummary.dll *The Land Cover summarizer model element*
LandCoverTypes.xml *Land Cover type definitions for the Land Cover Summary*
mwBestManagementPractice.dll *Best Management Practices model element*
mwChangeLandCover.dll *Land Cover Changer model element*
mwClimateChanger.dll *Climate Changer model element*
mwPopulationChanger.dll *Population Changer model element*
mwRainDataFilter.dll *Rain Data Filter data element*
ReservoirStorage.dll *Reservoir Storage editor model element*
ReservoirStorage.pdb *Program debug database for the above*

Macroinvertibrate Data Viewer (MIVViewer.msm)

(Installation\MacroInvert\Merge Module\mwMIVViewer\mwMIVViewer.sln)

This is the merge module for the Macrointertibrate Data Viewer MapWindow plugin. This consists primarily of the DLL for this plugin and a few dependencies. The files are placed in C:\Program Files\MapWindow\Plugins\mwMIVViewer.

Files Included

AxInterop.MSChart20Lib.dll *.NET wrapper for Microsoft Chart ActiveX control*
Interop.MSChart20Lib.dll *.NET wrapper for Microsoft Chart ActiveX control*
Mschrt20.msm *Merge module containing MS Chart control and dependencies*
mwMacroInvertDataViewer.dll *MacroInvertibrate Data Viewer plugin*

Predicted Habitat Simulation 1D Viewer (PHabSim1DViewer.msm)

(Installation\mwPhabSim1DViewer\MergeModule\mwPhabSim1DViewer\mwPhabSim1DViewer.sln)

This merge module contains the PHabSim 1D data viewer, as well as its help file and a few dependencies of this component. Files are installed to C:\Program Files\MapWindow\Plugins\mwPhabSim1DViewer.

Files Included

AxInterop.PE3DO32BLib.dll *.NET wrapper for ProEssentials controls*
Interop.PE3DO32BLib.dll *.NET wrapper for ProEssentials controls*
BLOBManage.dll *Custom DLL used to manage the data for this plugin*

Interop.BLOBMANAGELib.dll *.NET wrapper for custom BLOB dll*
Interop.PE3DO32BLib.dll *.NET wrapper for ProEssentials controls*
Interop.PESGO32BLib.dll *.NET wrapper for ProEssentials controls*
mwPhabSim1DViewer.dll *The PHabSim 1D MapWindow Plugin DLL*
Pe3do32b.ocx *Gigasoft ProEssentials conrol*
Pego32b.ocx *Gigasoft ProEssentials conrol*
Pegrp32b.dll *Gigasoft ProEssentials conrol*
Pepco32b.ocx *Gigasoft ProEssentials conrol*
Pepso32b.ocx *Gigasoft ProEssentials conrol*
Pesgo32b.ocx *Gigasoft ProEssentials conrol*
PHab1DUserManual.doc *User's Documentation for the PHabSim 1D Viewer*

Predicted Habitat Simulation 2D Viewer (PS2DViewer.msm)

(Installation\mwPS2DViewer\Merge Module\mwPS2DViewer\mwPS2DViewer.sln)

This merge module contains the PHabSim 2D viewer plugin as well as mapping components, a media player, and graphing components. Files are installed to C:\Program Files\MapWindow\Plugins\mwPS2DViewer.

Files Included
AxInterop.MapWinGIS.dll *.NET Wrapper for MapWindow Map component*
AxInterop.MediaPlayer.dll *.NET Wrapper for Windows Media Player component*
AxInterop.PESGO32BLib.dll *.NET wrapper for Proessentials graph control*
Interop.MediaPlayer.dll *.NET wrapper for Windows Media Player component*
Interop.PESGO32BLib.dll *.NET wrapper for Proessentials Graph Control*
mwPhabSim2DViewer.dll *PHabSim 2D Viewer MapWindow plugin component*
Pe3do32b.ocx *Gigasoft ProEssentials graphing component*
Pego32b.ocx *Gigasoft ProEssentials graphing component*
Pegrp32b.dll *Gigasoft ProEssentials graphing component*
Pepco32b.ocx *Gigasoft ProEssentials graphing component*
Pepso32b.ocx *Gigasoft ProEssentials graphing component*
Pesgo32b.ocx *Gigasoft ProEssentials graphing component*

Photo Viewer (mwPhotoViewer.msm)

(Installation\Photo Viewer\Merge Module\mwPhotoViewer\mwPhotoViewer.sln)
(InstallShield Wrapper at Installation\Photo Viewer\Standalone Plugin and Data Install\NooksackPhotoViewer.ise)

This merge module contains the Photo Viewer two DLL files; a very simple merge module. The files are installed to C:\Program Files\MapWindow\Plugins\mwPhotoViewer.

Files Included
AxInterop.MapWinGIS.dll *.NET wrapper for MapWindow map component*
Interop.MapWinGIS.dll *.NET wrapper for MapWindow map component*
mwPhotoViewer.dll *PhotoViewer MapWindow Plugin DLL*

Time Series Data Analyst (mwTimeSeries.msm)

(Installation\TimeSeries\MergeModule\mwTimeSeries\mwTimeSeries.sln)
(InstallShield Wrapper with Sample Data at Installation\TimeSeries\TimeSeriesDataAnalyst.ise)

This merge module contains the Time Series Analyst tool, formerly the Water Quality Analyst and Streamflow Analyst. The merge module contains the MapWindow plugin and a bunch of graphing components. The files are installed in C:\Program Files\MapWindow\Plugins\mwTimeSeries.

Files Included

AxInterop.MSFlexGridLib.dll *.NET wrapper for MS FlexGrid control*
AxInterop.PEGOALib.dll *.NET wrapper for Proessentials Controls*
AxInterop.PESGOALib.dll *.NET wrapper for Proessentials Controls*
Interop.MSFlexGridLib.DLL *.NET wrapper for MS FlexGrid control*
Interop.PEGOALib.dll *.NET wrapper for Proessentials controls*
Interop.PESGOALib.dll *.NET wrapper for Proessentials controls*
Msflxgrd.msm *Merge module containing Microsoft FlexGrid control.*
MDAC26.msm *Microsoft Data Access Components – for database access.*
mwTimeSeries.dll *The Time Series Analyst MapWindow plugin.*
Pe3do32a.ocx *Gigasoft ProEssentials graphing component.*
Pego32a.ocx *Gigasoft ProEssentials graphing component.*
Pegrp32a.dll *Gigasoft ProEssentials graphing component.*
Pepco32a.ocx *Gigasoft ProEssentials graphing component.*
Pepso32a.ocx *Gigasoft ProEssentials graphing component.*
Pesgo32a.ocx *Gigasoft ProEssentials graphing component.*

Watershed Characterization Report (mwWatershedChar.msm)

(Installation\Watershed Characterization\Merge Module\mwWatershedChar\mwWatershedChar.ise)
(InstallShield Wrapper at Installation\Watershed Characterization\Merge Module\WatershedChar.ise)

This merge module contains the Watershed Characterization MapWindow plugin as well as a large number of prerequisites. Most of the prerequisites are installed either to the windows system directory, or to the .NET assembly location (usually C:\Windows\Microsoft.NET\...). The plugin itself is installed to C:\Program Files\MapWindow\Plugins\mwWatershedChar.

Files Included

ADODB.dll *Active Data Objects database driver DLL*
AxInterop.MapWinGIS.dll *.NET Wrapper for MapWinGIS component*
AxInterop.MSChart20Lib.dll *.NET wrapper for Chart Control*
AxInterop.MSComCtl2.dll *.NET wrapper for Common Controls*
AxInterop.MSFlexGridLib.dll *.NET wrapper for MS Flexgrid Control*
AxInterop.PE3DOALib.dll *.NET wrapper for Gigasoft Proessentials*

AxInterop.PEGOALib.dll *.NET wrapper for Gigasoft Proessentials*
AxInterop.PEPCOALib.dll *.NET wrapper for Gigasoft Proessentials*
AxInterop.PESGOALib.dll *.NET wrapper for Gigasoft Proessentials*
Interop.MapWinGIS.dll *.NET wrapper for MapWinGIS component*
Interop.MSChart20Lib.dll *.NET wrapper for Chart Control*
Interop.MSComCtl2.dll *. NET wrapper for Common Controls*
Interop.MSFlexGridLib.dll *.NET wrapper for MS Flexgrid Control*
Interop.PE3DOALib.dll *.NET wrapper for Gigasoft Proessentials*
Interop.PEGOALib.dll *.NET wrapper for Gigasoft Proessentials*
Interop.PEPCOALib.dll *.NET wrapper for Gigasoft Proessentials*
Interop.PESGOALib.dll *.NET wrapper for Gigasoft Proessentials*
LifestagePlotter.dll *Fish Periodicity Lifestage plot control*
LifestagePlotter.pdb *Debug database for above control*
mwWatershedChar.chm *Help document for Watershed Characterization*
mwWatershedChar.dll *Watershed CharacterizationReport Generator Plugin*
mwWatershedChar.pdb *Debug database for above DLL*
ReportViewer.exe *Watershed Characterization Report Viewer stub*
ReportViewer.pdb *Debug database for above EXE*
rptViewerPass.dat *Data file for communication between report plugin and EXE*
Mschrt20.msm *Microsoft Chart component merge module*
Mscomct2.msm *Microsoft Common Controls 2 merge module*
Msdatasrc.dll *Microsoft DataSource control*
Msflxgrd.msm *Microsoft Flexgrid Control merge module*
Sqldmo.dll *Microsoft SQL Server Distributed Management Object*

Well Log Data Viewer (mwWellviewer.msm)

(Installation\Well Viewer\Merge Module\mwWellviewer\mwWellViewer.sln)
(InstallShield Wrapper at Installation\Well Viewer\Merge Module\WellViewer.ise)

This is a merge module to install the Well Log Data Viewer and dependencies other than MapWindow. Files are installed to C:\Program Files\MapWindow\Plugins\mwWellViewer.

Files Included
AxInterop.PESGOALib.dll *.NET Wrapper for Gigasoft ProEssentials control*
AxInterop.PESGOALib.dll *.NET Wrapper for Gigasoft ProEssentials control*
Interop.PEGOALib.dll *.NET Wrapper for Gigasoft ProEssentials control*
Interop.PESGOALib.dll *.NET Wrapper for Gigasoft ProEssentials control*
mwWellViewer.dll *Well Log Viewer Plugin for MapWindow*

Crystal Reports 8.5 Runtime Components (CrystalReports85.msm)

This merge module is produced and distributed by Business Objects, Inc. This provides the core redistributable components needed by software using Crystal Reports.

Crystal Reports .NET Data Access (Crystal_Database_Access2003.msm)

This merge module is produced and distributed by Business Objects, Inc. This provides the redistributables to allow Crystal Reports to connect to and use a Microsoft Access database.

Crystal Reports .NET English (Crystal_Database_Access2003_enu.msm)

This merge module is produced and distributed by Business Objects, Inc. This provides the core redistributable components which are specific to the US English language.

Crystal Reports .NET Managed Code (Crystal_Managed2003.msm)

This merge module is produced and distributed by Business Objects, Inc. This provides the core redistributable components needed by software using the Crystal extensions to Microsoft's .NET languages.

Seagate Crystal Reports Keycode Manager (Crystal_regwiz2003.msm)

This merge module is produced and distributed by Business Objects, Inc. This provides the registration key indicating which developer produced the software, and authorizing the client computer to use the Crystal Reports components.

Data Access Objects 3.60 (DAO360.msm)

This merge module is produced by Microsoft Corp. It provides the Data Access Objects SDK (Software Development Kit) used to access databases from code.

Distributed Component Object Model 95 (DCOM95.msm)

This merge module is produced by Microsoft Corp. This is a core requirement for most Windows applications.

Microsoft Foundations Classes 6.0 Libraries (MFC42.msm)

This merge module is produced by Microsoft Corp. This provides libraries and dependencies needed by software developed with Microsoft Visual C++.

Microsoft C Runtime Library 6.0 (MSVCRT.msm)

This merge module is produced by Microsoft Corp. This provides libraries and dependencies needed by software developed with Microsoft Visual C.

Microsoft C++ Runtime Library 6.0 (MSVCP60.msm)

This merge module is produced by Microsoft Corp. This provides libraries and dependencies needed by software developed with Microsoft Visual C++ 6.0.

Microsoft Chart Control 6.0 (MSCHRT20.MSM)

This merge module is produced by Microsoft Corp. This provides the Microsoft Chart control, a simple bar-chart style control.

Microsoft Chart VB Control (VB_Control_mschart_RTL_X86_---.msm)

This merge module is produced by Microsoft Corp. This provides the Microsoft Chart control, a simple bar-chart style control, packaged specifically for use with Visual Basic.

Microsoft Common Dialog Control 6.0 (COMDLG32.msm)

This merge module is produced by Microsoft Corp. This provides the DLL used to create "common dialogs" such as Print, Save As, Open and such.

Microsoft Component Category Manager Library (COMCAT.msm)

This merge module is produced by Microsoft Corp. This is a core requirement for most Windows applications that use ActiveX.

Microsoft Data Access Components (MDAC) 2.5 (MDAC25.msm)

This merge module is produced by Microsoft Corp. It provides the Data Access components, specifically the ODBC (Open Database Compatibility) interface and a bunch of drivers for interacting with various database formats. This is version 2.5 of the components.

MDAC 2.6 (MDAC26.msm)

This merge module is produced by Microsoft Corp. It provides the Data Access components, specifically the ODBC (Open Database Compatibility) interface and a bunch of drivers for interacting with various database formats. This is version 2.6 of the components.

MDAC 2.7 (MDAC27enu.msm)

This merge module is produced by Microsoft Corp. It provides the Data Access components, specifically the ODBC (Open Database Compatibility) interface and a bunch of drivers for interacting with various database formats. This is version 2.7 (US English) of the components.

Microsoft FlexGrid Control 6.0 (MSFLXGRD.msm)

This merge module is produced by Microsoft Corp. It provides the FlexGrid control, a grid control used to create a spreadsheet-style grid layout.

Microsoft OLE 2.40 for 95/NT4.0 (OLEAUT32.msm)

This merge module is produced by Microsoft Corp. This is a core requirement for most Windows applications that use ActiveX.

Microsoft Typelib Information Library (tlbinf32.msm)

This merge module is produced by Microsoft Corp. This is a core requirement for most Windows applications that use ActiveX.

Microsoft Windows Common Controls 6.0 (MSCOMCTL.msm)

This merge module is produced by Microsoft Corp. It provides the FlexGrid control, a grid control used to create a spreadsheet-style grid layout.

Microsoft Windows Common Controls-2 6.0 (MSCOMCT2.msm)

This merge module is produced by Microsoft Corp. It provides a large collection of controls which are commonly used, including animation controls and picture-related controls.

OLE Database Access 2.1 (OLEDB21.msm)

This merge module is produced by Microsoft Corp. It provides the OLE database access SDK for accessing and communicating with databases.

SQL Distributed Management Object (SQL-DMO.msm)

This merge module is produced by Microsoft Corp. It provides a convenient library to interact with Microsoft SQL Servers, performing common administration, backup, and query tasks.

Technical Documentation: 1D Habitat Viewer

Last Revision: 01/4/08

Table of Contents

1. *Table Design*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Modules and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*
   - *Project Files*
   - *Reference Settings*

1. Table Design

Following is a list of tables that must be included with the 1D Habitat Viewer Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters. *NOTE: This database is created (except for the tblPDFs and tblPhotographs tables) by exporting the data from the PHABWin-2002 program. Most of this data is in a specialized format that can only be written using this program.*

Table: tblCalibrationSetPoints
Description: Contains the Velocity and Manning's N values for each point in the given Calibration Set.
NOTE: This table is created bye PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	Unique ID for each Calibration Set point.
CalSetID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Calibration Set that this point belongs to. It corresponds with the *DBCode* Field in the *tblCalibrationSets* table.
Point 2	Number	Single	No	No	Velocity at .2 depth for this calibration set point.
Point 8	Number	Single	No	No	Velocity at .8 depth for this calibration set point.
ManN	Number	Single	No	No	Manning's N value for this calibration set point.

Table: tblCalibrationSets
Description: Contains the Water Surface Level (WSL) and StageQ values for each Calibration Set at a given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Calibration Set. It corresponds with the *CalSetID* Field in the *tblCalibrationSetPoints* table.

XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this Calibration Set belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.
ID	Number	Long Integer	Yes (Duplicates OK)	No	This is an ID for this Calibration Set that is unique for the given Cross Section. *NOTE: This value can be duplicated for each unique Cross Section, but it cannot be duplicated within the same Cross Section.*
WSL	Number	Single	No	No	This is the Water Surface Level (WSL) value for this Calibration Set.
StageQ	Number	Singe	No	No	This is the StageQ that the WSL value represents for this Calibration Set.

Table: tblCrossSectionPoints
Description: Contains the Point Location (X, Y, and Z values) and the Channel Index for each point in the given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Cross Section point.
XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this point belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.
X	Number	Single	No	No	This is the X-value of the location of this cross sectional point.
Y	Number	Single	No	No	This is the Y-value of the location of this cross sectional point.
Z	Number	Single	No	No	This is the Z-value of the location of this cross sectional point.
Channel Index	Number	Single	No	No	This is the Channel Index for this cross sectional point.

Table: tblCrossSections
Description: Contains the Cross Section ID, Thalweg, and Stage Zero Flow (SZF) values for each Cross Section in the given Result Set.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Cross Section. It corresponds with the *XSecID* Field in the *tblCalibrationSets* table, the *tblCrossSectionPoints* table, the *tblResultsHabefQ1Q2* table, the *tblResultsHabefSpeciesCompare* table, the *tblResultsHabtae* table, the *tblResultsVelocity* table, and the *tblResultsWSL* table.
ResultID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Result Set that this Cross Section belongs to. It corresponds with the *DBCode* Field in the *tblResultSets* table.
ID	Number	Single	Yes (Duplicates OK)	No	This is an ID for this Cross Section that is unique for the given Result Set. *NOTE: This value can be duplicated for each unique Result Set, but it cannot be duplicated within the same Result Set.*
Thalweg	Number	Single	No	No	This is the Thalweg value for this Cross Section.
SZF	Number	Single	No	No	This is the Stage Zero Flow (SZF) value for this Cross Section.

Table: tblPDFs **Description:** Contains the PDF Documents that are available for the given Result Set (not always created).

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each PDF Document.
ResultID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Result Set that this PDF Document belongs to. It corresponds with the *DBCode* Field in the *tblResultSets* table.

PDF_Filename	Text	255	No	No	This is the File path where the PDF Document is located. *NOTE: This value is a relative path value. It is stored relative to this database.* *For Example:* *\Site_Name\PDFs\Filename.PDF*

Table: tblPhotographs
Description: Contains the Photographs, and their comments, that are available for the given Result Set (not always created).

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Photograph.
ResultID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Result Set that this Photograph belongs to. It corresponds with the *DBCode* Field in the *tblResultSets* table.
Photo_Filename	Text	255	No	No	This is the File path where the Photograph is located. *NOTE: This value is a relative path value. It is stored relative to this database.* *For Example:* *\Site_Name\Photos\Filename.jpg*
Comments	Memo		No	No	These are the comments about this Photograph. These are displayed in the Viewer for the user to see.

Table: tblProductionFlows
Description: Contains the simulated Production Flows for the given Result Set.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each simulated Production Flow.

ResultID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Result Set that this simulated Production Flow belongs to. It corresponds with the *DBCode* Field in the *tblResultSets* table.
Flow	Number	Single	No	No	This **is** the simulated Production Flow value.

Table: tblResultSet
Description: Contains the available Result Sets exported for the given Station.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Result Set. It corresponds with the *ResultID* Field in the *tblCrossSections* table, the *tblPDFs* table, the *tblPhotographs* table, the *tblProductionFlows* table, and the *tblSpecies* table.
StationID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Result Set belongs to. It corresponds with the *DBCode* Field in the *tblStations* table.
Name	Text	50	No	No	This is the name given to this Result Set.

Table: tblResultsHabefQ1Q2
Description: Contains the simulated Effective Habitat Flow Comparison Results for the given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of simulated Effective Habitat Flow Comparison Results.
XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this set of simulated Effective Habitat Flow Comparison Results belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.

HABBloc	OLE Object		No	No	This is the set of simulated Effective Habitat Flow Comparison Results. *NOTE: This value is in a Binary Large OBject (BLOB) format. It is specially written by the PHABWin-2002 program.*

Table: tblResultsHabefSpeciesCompare
Description: Contains the simulated Effective Habitat Species Comparison Results for the given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of simulated Effective Habitat Species Comparison Results.
XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this set of simulated Effective Habitat Species Comparison Results belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.
HABBloc	OLE Object		No	No	This is the set of simulated Effective Habitat Species Comparison Results. *NOTE: This value is in a Binary Large OBject (BLOB) format. It is specially written by the PHABWin-2002 program.*

Table: tblResultsHabtae
Description: Contains the simulated Habitat Results for the given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of simulated Habitat Results.
XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this set of simulated Habitat Results belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.

HABBloc	OLE Object		No	No	This is the set of simulated Habitat Results. *NOTE: This value is in a Binary Large OBject (BLOB) format. It is specially written by the PHABWin-2002 program.*

Table: tblResultsVelocity
Description: Contains the simulated Velocity Results for the given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of simulated Velocity Results.
XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this set of simulated Velocity Results belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.
VELBloc	OLE Object		No	No	This is the set of simulated Velocity Results. *NOTE: This value is in a Binary Large OBject (BLOB) format. It is specially written by the PHABWin-2002 program.*

Table: tblResultsWSL
Description: Contains the simulated Water Surface Level (WSL) Results for the given Cross Section.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of simulated Water Surface Level (WSL) Results.
XSecID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Cross Section that this set of simulated Water Surface Level (WSL) Results belongs to. It corresponds with the *DBCode* Field in the *tblCrossSections* table.

WSLBloc	OLE Object		No	No	This is **the** set of simulated Water Surface Level (WSL) Results. *NOTE: This value is in a Binary Large Object (BLOB) format. It is specially written by the PHABWin-2002 program.*

Table: tblSpecies
Description: Contains the Information for each Species and Lifestage pair available in the given Result Set.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Species and Lifestage pair. It corresponds with the *SpeciesID* Field in the *tblSpeciesSI* table.
ResultID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Result Set that this Species and Lifestage pair belongs to. It corresponds with the *DBCode* Field in the *tblResultSets* table.
Species	Text	255	No	No	This is the name of the Species.
Lifestage	Text	255	No	No	This is the name of the Lifestage of the Species.

Table: tblSpeciesSI
Description: Contains the simulated Suitability Index (SI) values (Depth, Velocity, Temperature, and Channel Index) for the given Fish.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each simulated Suitability Index (SI) value.
SpeciesID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Fish that this simulated Suitability Index (SI) value belongs to. It corresponds with the *DBCode* Field in the *tblSpecies* table.
SI Type	Number	Long Integer	No	No	This is the Suitability Index (SI) Type for this value. It corresponds with the *DBCode* Field in the *tblSpeciesSIType* table.

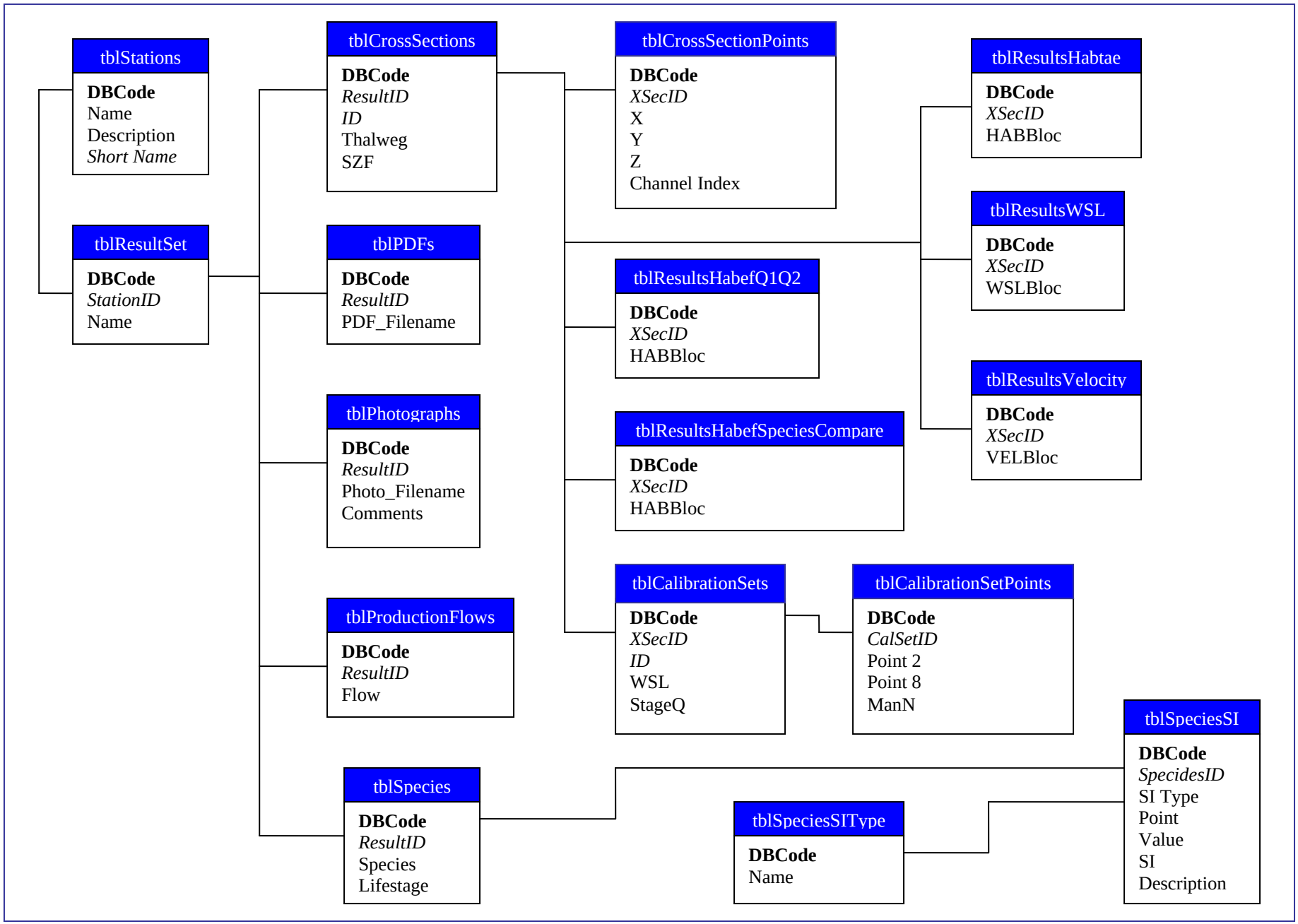
Point	Number	Long Integer	No	No	This is the Cross Section Point number for this value (keeps the data ordered).
Value	Number	Single	No	No	This is the Depth, Velocity, Temperature, or Channel Index value.
SI	Number	Single	No	No	This is the Suitability Index (SI) value for the given value.
Description	Text	255	No	No	This is the Description of the Suitability Index (SI) value. *NOTE: This value is only used for the Channel Index plot type.*

Table: tblSpeciesSIType
Description: Contains the available Species Suitability Index (SI) Plot Types.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Species Suitability Index (SI) Plot Type. It corresponds with the *SI Type* Field in the *tblSpeciesSI* table.
Name	Text	50	No	No	This is the Name of the Type of the Species Suitability Index (SI) Plot Type. *NOTE: The available Plot Types should be: Depth, Velocity, Temperature, and Channel Index.*

Table: tblStations
Description: Contains the information (ID, Name, and Description) for each Station.
NOTE: This table is created by PHABWin-2002 program.

Field	Type	Size	Key Field	Primary Key	Comments
DBCode	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Station. It corresponds with the *StationID* Field in the *tblResultSet* table.
Name	Text	255	No	No	This is the Name of the Station.
Description	Text	255	No	No	This is the Description of the Station.
Short Name	Text	255	Yes (Duplicates OK)	No	This is the Short Name for the Station.


tblStations
DBCode
Name
Description
Short Name
tblResultSet
DBCode
StationID
Name
tblCrossSections
DBCode
ResultID
ID
Thalweg
SZF
tblPDFs
DBCode
ResultID
PDF_Filename
tblPhotographs
DBCode
ResultID
Photo_Filename
Comments
tblProductionFlows
DBCode
ResultID
Flow
tblSpecies
DBCode
ResultID
Species
Lifestage
tblCrossSectionPoints
DBCode
XSecID
X
Y
Z
Channel Index
tblResultsHabefQ1Q2
DBCode
XSecID
HABBloc
tblResultsHabefSpeciesCompare
DBCode
XSecID
HABBloc
tblCalibrationSets
DBCode
XSecID
ID
WSL
StageQ
tblCalibrationSetPoints
DBCode
CalSetID
Point 2
Point 8
ManN
tblResultsHabtae
DBCode
XSecID
HABBloc
tblResultsWSL
DBCode
XSecID
WSLBloc
tblResultsVelocity
DBCode
XSecID
VELBloc
tblSpeciesSIType
DBCode
Name
tblSpeciesSI
DBCode
SpecidesID
SI Type
Point
Value
SI
Description

2. Data Needs

There are three types of data needed for the1D Habitat Viewer: a point shapefile, a database type, and a database.

- Point Shapefile – this needs to be in the same projection as the other shapefiles in your project. It contains the point locations of where each of the Stations are located. This file will have a tag that will be associated with it by the plug-in: "rapidlocations". There needs to be a field containing the *Station Name* in this shapefile.
- Database Type – this is the type of database that you are connecting to. IE: Access, SQL Server.
- Database – this is the database of 1D Habitat Data that has been exported from the PHABWin-2002 Program, and had the PDF and Photograph tables added. This database needs to be organized as described in *Section 1. Table Design.*

The layer corresponding with the point shapefile stated above is marked with the tag "rapidlocations ". If the user removes the layer associated with this tag without first closing the 1D Habitat Viewer plug-in, then the Shapefile settings will be reset, and if the Project is then saved, then the Shapefile settings in the project file will also be over-written.

As stated above, the shapefile requires one field to be found in the data table. Following is the description of the required Field in the shapefile (an example field name, the field type, a description of the data that gets entered into this field, and how the field data is associated with the data in the database.

Required data layer field:

a. *Station Name* – String – The Name for the Station at this location. It corresponds with the DBCode field in the tblStations table found in the database. Each value that is found in this field (in the shapefile) also needs to be found in the database.

3. Dependencies

The 1D Habitat Viewer plug-in requires the following software components and modules to be installed:

Software:

Gigasoft's ProEssentials Graphing Tools Version 4
MapWindow 3.1
Visual Studio .NET 2003

Modules and Components:
BLOBManage 1.0 Type Library
Gigasoft ProEssentials 3D Sci-Graph v4
Gigasoft ProEssentials Sci-Graph v4
MapWindow Interfaces
MapWinGIS Map Control

Integrated Development Environment (IDE)
Visual Studio .NET 2003

4. Setup

To begin using the 1D Habitat Viewer, it must first be loaded into MapWindow. Once it is loaded, click on the 1D Habitat icon, , located on the Map Window toolbar If the shapefile (if associated) and database associations (as described in *Section 2: Data Needs*) have been configured properly and the database is accessible, then the 1D Habitat Viewer will appear allowing the user to select and view the 1D Habitat Simulation data for available stations.

If the data associations have not been properly set, then a Connection Form will appear allowing you to set the shapefile (if desired), the database type, and the database. Then, after the shapefile (if desired), the database type, and the database have been properly associated, select the 1D Habitat icon, , from the Map Window toolbar and the viewer will appear allowing the user to view the data.

See the User's Manual for more information and details on associating the data with the 1D Habitat Viewer.

5. Code Compiling

Compiling the 1D Habitat Viewer is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 3: Dependencies* are present, load the project into Visual Studio .Net 2003. This Plugin was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsDBFunctions.vb	Contains functions used for accessing and updating the associated database.
clsEHFlowInfo.vb	Contains classes for accessing and storing Expected Habitat data.

clsObservedInfo.vb	Contains a class for accessing and storing Calibration Set (Observed) Data.
clsObservedVelocities.vb	Contains a class for accessing and storing Observed Velocity data for the Calibration Set (Observed) Data.
clsPredictedFlow.vb	Contains a class for accessing and storing Flow, Water Surface Level (WSL), and Velocity data for the simulated Predicted Flow Data.
clsPredictedHabitat.vb	Contains a class for access and storing Predicted Habitat data for the simulated Predicted Flow Data.
clsPS1DMain.vb	Contains a class that implements the MapWindow plugin interface.
clsSpecies.vb	Contains a class for accessing and storing Species and SpeciesSI data for the simulated Predicted Flow Data.
clsXSecPoints.vb	Contains a class for accessing and storing Cross Section Data.
frmDBConnection.vb, frmDBConnection.resx	Displays the Connection form that allows the user to associate the shapefile and shapefile field (if desired), the database type, and the database with the 1D Habitat Viewer plug-in.
frmHotSpot.vb, frmHotSpot.resx	Displays the form that allows the user view the data for a specific point on any of the graphs.
frmPS1DVisualization.vb, frmPS1DVisualization.resx	Displays the 1D Habitat Viewer form. This form contains the many plots, photographs, and other miscellaneous viewable data for available stations. You show this form by selecting the 1D Habitat icon, , from the Map Window toolbar or by selecting one or more points on the associated shapefile.
frmSelectLayer.vb, frmSelectLayer.resx	Displays the form to select the point shapefile associated with the 1D Habitat Viewer plug-in. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk. This form will only be shown if the user decides to associate a shapefile with this plug-in.
frmSelectPDF.vb, frmSelectPDF.resx	Displays the form to select a PDF Filename to view.
ImageConverter.vb	Contains a class that implements functions that allows you to convert images to and from an IPictureDisp object. This is needed so that the VBCompatibility.dll does not need to be referenced. This class allows the 1D Habitat image, , to be associated with the points on the associated (if desired) shapefile.
modDSSDefinitions.vb	This module contains the variables that define the table and field variable name for the 1D Habitat Viewer database. These variables are used throughout the project, so if anything should change in the database, the table or field name value only has to be changed in 1 location in the program.
modFormFunctionality.vb	This module contains variables for copying and saving the data in the various plots in the 1D Habitat Viewer.

modGlobals.vb	This module contains the variables used throughout the forms, such as the form declaration variables, MapWindow variables, and others.
modPlotColors.vb	This module contains the variable definitions and functions for creating and setting the various plots' colors.
modPlotNameDefinitions.vb	This module contains the variable definitions for the available plot types in the 1D Habitat Viewer.
modUtils.vb	This module contains functions that are used throughout the project for reporting errors, file functionality, searching shapefiles, and other necessary functionality.
PluginInfo.vb	Contains a class that implements an interface to access (read from and write to) the Project File.
trout1D.bmp	Bitmap version of the trout1D.ico. It is used as a custom image for the point shapefile associated with the 1D Habitat Viewer plug-in. *NOTE: this file needs to be an embedded resource*
trout1D.ico	Icon that is used as the Map Window Legend picture when using a custom image for the point shapefile if associated with the 1D Habitat Viewer. It also is the icon on the Map Window Toolbar for the 1D Habitat Viewer plug-in. *NOTE: this file needs to be an embedded resource*

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

AxPE3DO32BLib	CopyLocal = True
AxPESGO32BLib	CopyLocal = True
BLOBMANAGELib	CopyLocal = True
MapWinGIS	CopyLocal = False
MapWinInterfaces	CopyLocal = False
PE3DO32BLib	CopyLocal = True
PESGO32BLib	CopyLocal = True
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwPhabSim1DViewer.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwPhabSim1DViewer* subdirectory in the *Plugins* folder. Next time that MapWindow is run, if the mwPhabSim1DViewer.dll was built to the correct folder, the updated changes to the 1D Habitat Viewer will be available.

Technical Documentation: 2D Habitat Viewer

Last Revision: 06/13/06

Table of Contents

1. *Table Design*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Modules and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*
   - *Project Files*
   - *Reference Settings*

1. Table Design

Following is a list of tables that must be included with the 2D Habitat Viewer Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling, and cases, and types for each table and its parameters.
NOTE: Some of the data in this database, and some of the files used by the 2D Habitat Viewer plug-in were created using a specialized 2D Data Importer. The data created using this importer is in a specialized format that can only be written using this program.

Table: tblBackgroundImageFiles
Description: Contains the available Background Image File name and location for a given station.

Field	Type	Size	Key Field	Primary Key	Comments
Background_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	Unique ID for each Background Image.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Station that this Background Image belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.
Image_Filename	Text	255	No	No	This is the File path where the Background Image is located. This file must be in a BMP, PNG, or GIF format. If this Background Image has a World File associated with it, the World File needs to be located next to the Background Image File. *NOTE: This value is a relative path value. It is stored relative to this database.* *For Example: \Station\BackgroundImage\Filename.bmp.*

Table: tblFishDistribution
Description: Contains the available Fish Distribution File names, types, and locations for each Species.
NOTE: this set of data is not based on Station, it is Area wide.

Field	Type	Size	Key Field	Primary Key	Comments
FD_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Fish Distribution File.
Species_Name	Text	255	No	No	This is the Name of the Species that this Fish Distribution File belongs to. *NOTE: if this file is a background file, then the value* = ALL.
FD_FileType	Text	255	No	No	This is the Type of Shapefile or Grid that this Fish Distribution File is. *NOTE : Possible values include Fish Distribution, Hillshade, Watershed Boundary, County Boundary, or Marine Boundary.*
FD_Filename	Text	255	No	No	This is the File path where the Fish Distribution File is located. This file must be a valid Shapefile or Grid file. *NOTE: This value is a relative path value. It is stored relative to this database. For Example: \FishDistribution\FDLayers\Filename.shp*

Table: tblFishObsFiles
Description: Contains the available Fish Observation shapefile overlays for the given Station.

Field	Type	Size	Key Field	Primary Key	Comments
FishObs_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Fish Observation Overlay.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Fish Observation Overlay belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.

FishObs_Filename	Text	255	No	No	This is the File path where the Fish Observation Overlay File is located. This file must be a valid Polygon Shapefile. *NOTE: This value is a relative path value. It is stored relative to this database.* *For Example:* *\Station\FishObs\Filename.shp*

Table: tblFlows
Description: Contains the Flow values for the given Station.
NOTE: This table needs to be filled in previous to running the 2D Data Importer.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
Flow_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Flow. It corresponds with the *Flow_ID* Field in the *tblMeshFiles* table.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Flow belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.
Flow_cms	Number	Double (Fixed: 2 decimal places)	No	No	This is the Flow value in *cubic meters per second (*cms *:* m^3/sec*).*
Flow_cfs	Number	Double (Fixed: 2 decimal places)	No	No	This is the Flow value in *cubic feet per second (*cfs : ft^3/sec*).*
AVI_Frame	Number	Long Integer	No	No	This is the AVI Frame number that this Flow corresponds with.

Table: tblMediaFiles
Description: Contains the File paths and names for the available miscellaneous Media Files for a given Station.
NOTE: Available miscellaneous Media File Types are AVI, Aerial View, PDF, Photograph, and Habitat Suitability Curve Definition Files.

Field	Type	Size	Key Field	Primary Key	Comments
Media_Files_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Media File.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Media File belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.
Media_Type_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the Media Type for this Media File. It corresponds with the *Media_Type_ID* Field in the *tblMediaTypes* table.
Media_Filename	Text	255	No	No	This is the File path where the Media File is located. *NOTE: This value is a relative path value. It is stored relative to this database.* *For Example:* *\Station\Miscellaneous\Filename.ext*
Comments	Memo		No	No	This contains any information that is needed to be displayed or noted about this Media File.

Table: tblMediaTypes
Description: Contains the available miscellaneous Media File Types.
NOTE: This table should already be completed for you. Only these types are viewable.

Field	Type	Size	Key Field	Primary Key	Comments
Media_Type_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Media Type. It corresponds with the *Media_Type_ID* Field in the *tblMediaFiles* table.
Media_Type	Text	50	No	No	This is the name of the Media Type. *NOTE: Available Types = AVI, Aerial View, PDF, Photograph,*

					HSC
Description	Text	255	No	No	This is a description of the Media File Type. *For Example: For Media_Type =* AVI, *Description =* AVI Files

Table: tblMeshFiles
Description: Contains the Base Meshfile, and the compressed Flow and Species Files for a given Station.
NOTE: This table and the files associated with this table are populated and created by the 2D Data Importer program.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
Mesh_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Mesh File or Compressed File.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Mesh File or Compressed File belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.
Flow_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Flow that this Mesh File or Compressed File belongs to. It corresponds with the *Flow_ID* Field in the *tblFlows* table. *NOTE: This value is only used if this Mesh File is a Compressed Flow or Species File. Otherwise it's value = 0.*
Species_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Species that this Mesh File or Compressed File belongs to. It corresponds with the *Species_ID* Field in the *tblSpecies* table. *NOTE: This value is only used if this Mesh File is a Compressed Species File. Otherwise it's value = 0.*
Mesh_Filename	Text	255	No	No	This is the File path where the Mesh File or Compressed File is located. *NOTE: This value is a relative path value. It is stored relative to this database.*

					For Example: \Station\Mesh\FileType\Filename.ext

Table: tblSpecies
Description: Contains the available Species for a given Station.
NOTE: This table needs to be filled in previous to running the 2D Data Importer.
The values in this table are only used with the tblMeshFiles *table, they are not used for the* tblFishDistribution *table.*

Field	Type	Size	Key Field	Primary Key	Comments
Species_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Species. It corresponds with the *Species_ID* Field in the *tblMeshFiles* table.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Species belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.
Species_Name	Text	50	No	No	This is the name of the Species. *NOTE: The format for this value is: Species : Lifestage*

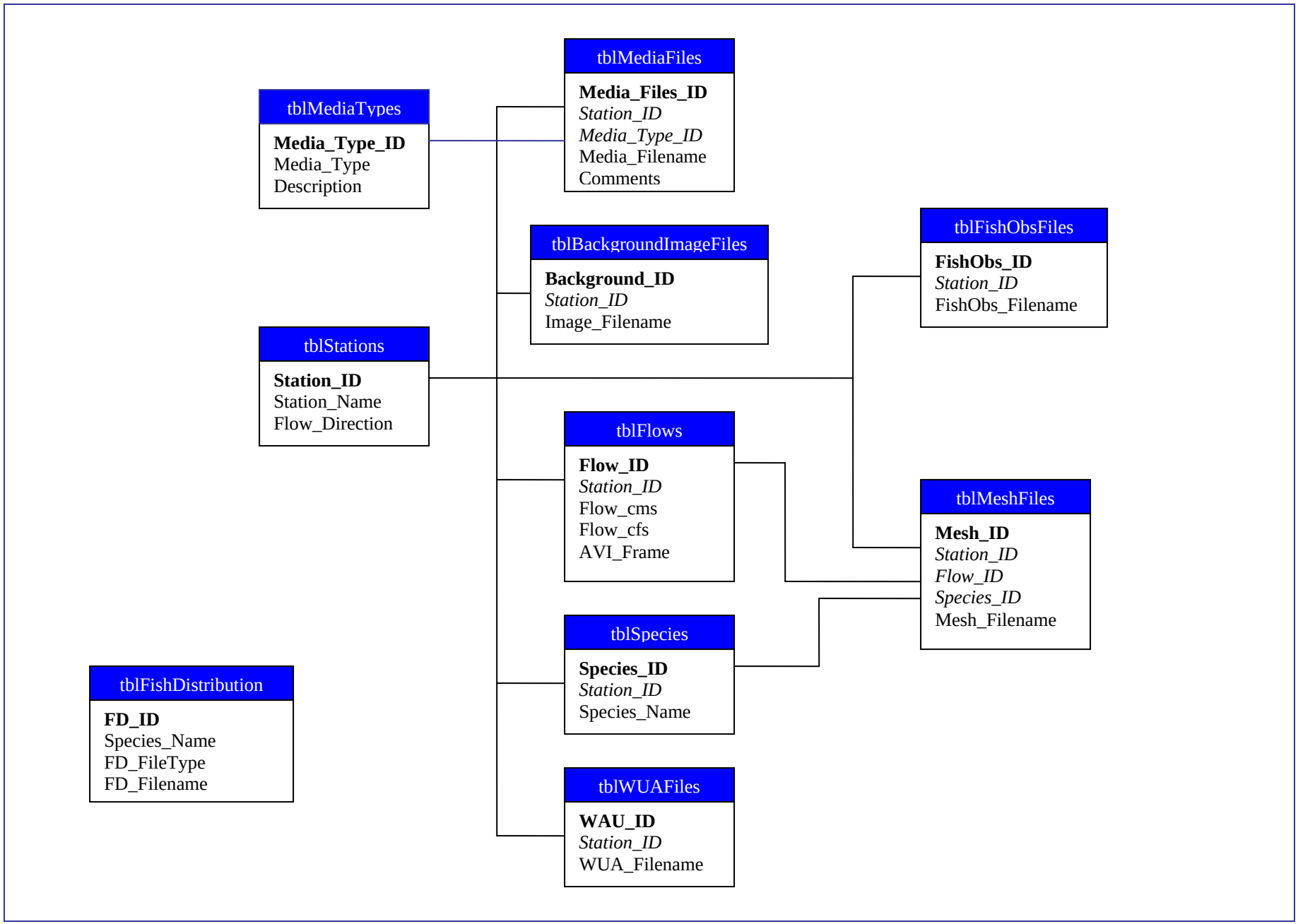
Table: tblStations
Description: Contains the available Stations and the Flow Direction for the river section at that Station.
NOTE: This table needs to be filled in previous to running the 2D Data Importer.

Field	Type	Size	Key Field	Primary Key	Comments
Station_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Station. It corresponds with the *Station_ID* Field in the *tblBackgroundImageFiles* table, the *tblFishObsFiles* table, the *tblFlows* table, the *tblMediaFiles* table, the *tblMeshFiles* table, the *tblSpecies* table, and the *tblWUAFiles* table.
Station_Name	Text	50	No	No	This is the Name of the Station.
Flow_Direction	Text	2	No	No	This is the Flow Direction for the river section at this Station. *NOTE: Valid values are the*

					standard 8 compass directions = (N, NE, E, SE, S, SW, W, NW).

Table: tblWUAFiles
Description: Contains the File paths and names for the available Weighted Usable Area (WUA) Files at the given Station.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
WUA_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Weighed Usable Area (WUA) File.
Station_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Station that this Weighted Usable Area (WUA) File belongs to. It corresponds with the *Station_ID* Field in the *tblStations* table.
WUA_Filename	Text	255	No	No	This is the File path where the Weighted Usable Area (WUA) File is located. *NOTE: This value is a relative path value. It is stored relative to this database.* *For Example:* *\Station\WUA\Filename.csv*


tblMediaFiles
Media_Files_ID
Station_ID
Media_Type_ID
Media_Filename
Comments
tblMediaTypes
Media_Type_ID
Media_Type
Description
tblBackgroundImageFiles
Background_ID
Station_ID
Image_Filename
tblFishObsFiles
FishObs_ID
Station_ID
FishObs_Filename
tblStations
Station_ID
Station_Name
Flow_Direction
tblFlows
Flow_ID
Station_ID
Flow_cms
Flow_cfs
AVI_Frame
tblMeshFiles
Mesh_ID
Station_ID
Flow_ID
Species_ID
Mesh_Filename
tblSpecies
Species_ID
Station_ID
Species_Name
tblFishDistribution
FD_ID
Species_Name
FD_FileType
FD_Filename
tblWUAFiles
WAU_ID
Station_ID
WUA_Filename

1271

2. Data Needs

There are two types of data needed for the 2D Habitat Viewer: a point shapefile and a database.

- Point Shapefile – this needs to be in the same projection as the other shapefiles in your project. It contains the point locations of where each of the Stations are located. This file will have a tag that will be associated with it by the plug-in: "2DIntensiveSites". There needs to be a field containing the *Station ID* in this shapefile.
- Database – this is the database of 2D Habitat Data that has been imported using the *2D Data Importer* and the rest of the tables created/entered manually. This database needs to be organized as described in *Section 1. Table Design.*

The layer corresponding with the point shapefile stated above is marked with the tag "2DIntensiveSites". If the user removes the layer associated with this tag without first closing the 2D Habitat Viewer plug-in, then the Shapefile settings will be reset, and if the Project is then saved, then the Shapefile Connection Settings in the project file will also be over-written.

As stated above, the shapefile requires one field to be found in the data table. Following is the description of the required Field in the shapefile (an example field name, the field type, a description of the data that gets entered into this field, and how the field data is associated with the data in the database.

Required data layer field:

a. *Station ID* – Integer – The ID for the Station at this location. It corresponds with the *Station_ID* field in the *tblStations* table found in the database. Each value that is found in this field (in the shapefile) also needs to be found in the database.

3. Dependencies

The 2D Habitat Viewer plug-in requires the following software components and modules to be installed:

Software:
Gigasoft's ProEssentials Graphing Tools Version 4
MapWindow 3.1
Windows Media Player
Visual Studio .NET 2003

Modules and Components:
Gigasoft ProEssentials Sci-Graph v4
ICSharpCode.SharpZipLib

MapWindow Interfaces
MapWinGIS Map Control

Integrated Development Environment (IDE)
Visual Studio .NET 2003

4. Setup

To begin using the 2D Habitat Viewer, it must first be loaded into MapWindow. Once it is loaded, click on the 2D Habitat icon, , located on the Map Window toolbar If the shapefile and database associations (as described in *Section 2: Data Needs*) have been configured properly and the database is accessible, then the 2D Habitat Viewer will appear allowing the user to select and view the 2D Habitat Data for available stations.

If the data associations have not been properly set, then a Connection Form will appear allowing you to set the shapefile and the database. Then, after the shapefile and the database have been properly associated, select the 2D Habitat icon, , from the Map Window toolbar and the viewer will appear allowing the user to view the data.

See the User's Manual for more information and details on associating the data with the 2D Habitat Viewer.

5. Code Compiling

Compiling the 2D Habitat Viewer is a fairly straightforward task. After ensuring that all of the required components (those discussed in *Section 3: Dependencies)* are present, load the project into Visual Studio .Net 2003. This Plugin was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsDirectionPtVals.vb	Contains a class that defines variables and functions used for drawing the Direction Arrow on the Map.
clsHSCData.vb	Contains classes for accessing and storing Habitat Suitability Curve Data.
clsMapData.vb	Contains a class that defines variables and functions for loading Shapefiles into the Map.
clsPS2DMain.vb	Contains a class that implements the MapWindow plugin interface.

clsWUAData.vb	Contains classes for accessing and storing Weighted Usable Area (WUA) Data.
frmDBConnection.vb, frmDBConnection.resx	Displays the Connection form that allows the user to associate the shapefile and shapefile field, and the database with the 2D Habitat Viewer plug-in.
frmFlowFrameIndex.vb, frmFlowFrameIndex.resx	Displays the form that allows the user view the AVI Frame vs. Flow Indexing.
frmPS2DViewer.vb, frmPS2DViewer.resx	Displays the 2D Habitat Viewer form. This form contains the Physical (Map), HSC, WUA, Fish Distribution, and other Miscellaneous viewable data for available stations. You show this form by selecting the 2D Habitat icon, , from the Map Window toolbar or by selecting one or more points on the associated shapefile.
frmSelectFishObs.vb, frmSelectFishObs.resx	Displays the form to select the Fish Observation Overlay File to view with the Physical (Map) Data.
frmSelectLayer.vb, frmSelectLayer.resx	Displays the form to select the point shapefile associated with the 2D Habitat Viewer plug-in. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk.
ImageConverter.vb	Contains a class that implements functions that allows you to convert images to and from an IPictureDisp object. This is needed so that the VBCompatibility.dll does not need to be referenced. This class allows the 2D Habitat image, , to be associated with the points on the associated shapefile.
modColorSchemes.vb	This module contains the variables that define the Coloring Schemes for the Physical (Map) Data Plots.
modDBDefinitions.vb	This module contains the variables that define the table and field variable name for the 2D Habitat Viewer database. These variables are used throughout the project, so if anything should change in the database, the table or field name value only has to be changed in 1 location in the program.
modFDDefinitions.vb	This module contains the variables that define the Coloring Schemes, Legend Colors, File Types, Fish Distribution Types, and Shapefile Info for the Fish Distribution Data.
modGlobals.vb	This module contains the variables used throughout the forms, such as the form declaration variables, MapWindow variables, conversion constants, resizing variables, Shapefile and Database variables, and others.
modHSCDefinitions.vb	This module contains the variable definitions for the Habitat Suitability Curve Data.
modMapDefinitions.vb	This module contains the variable definitions of the Plot Types, Shapefile Fieldnames, and Species SI Types for the Physical (Map) Data.
modOtherDefinitions.vb	This module contains the enumeration and variable

	definitions for the Miscellaneous Data.
modUtils.vb	This module contains functions that are used throughout the project for reporting errors, file functionality, Drawing the North and Direction Arrows, shapefile functionality, decompressing Physical Data files, conversion functionality, and other necessary functionality.
PluginInfo.vb	Contains a class that implements an interface to access (read from and write to) the Project File.
ScaleBarUtils.vb	Contains enumerations, conversion constants, and a class to draw a Scalebar in the desired units.
trout2D.bmp	Bitmap version of the trout2D.ico. It is used as a custom image for the point shapefile associated with the 2D Habitat Viewer plug-in. *NOTE: this file needs to be an embedded resource*
trout2D.ico	Icon that is used as the Map Window Legend picture when using a custom image for the point shapefile associated with the 2D Habitat Viewer. It also is the icon on the Map Window Toolbar for the 2D Habitat Viewer plug-in. *NOTE: this file needs to be an embedded resource*

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

AxMapWinGIS	CopyLocal = True
AxMediaPlayer	CopyLocal = True
AxPESGO32BLib	CopyLocal = True
ICSharpCode.SharpZipLib	CopyLocal = True
MapWinGIS (Interop.MapWinGIS)	CopyLocal = True
MapWinInterfaces	CopyLocal = False
MediaPlayer	CopyLocal = True
PESGO32BLib	CopyLocal = True
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwPhabsim2DViewer.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwPS2DViewer* subdirectory in the *\Plugins* folder. Next time that MapWindow is run, if the mwPhabsim2DViewer.dll was built to the correct folder, the updated changes to the 2D Habitat Viewer will be available.

Technical Documentation:
Best Management Practices Tool

Last Revision: Feb 21, 2006

Table of Contents

1. *Table Designs*
   - *Table Descriptions and Notes*

2. *GIS Data Needs*
   - *Location of GIS Data*

3. *Dependencies*
   - *Software*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Building*
   - *Code Compiling*
   - *Installations*

1. Table Designs

Best Management Practices Database: BestManagementPractices.bmp

Table: BMP_DefaultValues

Field Name	Data Type	Description
BMP_Name	Text	The name of the best management practice
PercentApplication	Long Integer	The percentage of the land cover or watershed area the best management practice is applied to
N_Reduction	Long Integer	The percentage of reduction for nitrogen
NH3_NH4_N_Reduction	Long Integer	The percentage of reduction for NH_3 + NH_4 - N
NO2_N03_N_Reduction	Long Integer	The percentage of reduction for NO_2 + NO_3 - N
Phos_Reduction	Long Integer	The percentage of reduction for phosphorus
BOD_Reduction	Long Integer	The percentage of reduction for BOD
FecalColBac_Reduction	Long Integer	The percentage of reduction for fecal coliform bacteria
11	Yes/No	Does the BMP affect this land cover classification (OpenWater)
12	Yes/No	Does the BMP affect this land cover classification (Perennial Ice/Snow)
21	Yes/No	Does the BMP affect this land cover classification (Low Intensity Residential)
22	Yes/No	Does the BMP affect this land cover classification (High Intensity Residential)
23	Yes/No	Does the BMP affect this land cover classification (Commercial/Industrial/Transportation)
31	Yes/No	Does the BMP affect this land cover classification (Bare Rock/Sand/Clay)
32	Yes/No	Does the BMP affect this land cover classification (Quarries/Strip Mines, Gravel Pits)
33	Yes/No	Does the BMP affect this land cover classification (Transitional)
41	Yes/No	Does the BMP affect this land cover classification (Deciduous Forest)

42	Yes/No	Does the BMP affect this land cover classification (Evergreen Forest)
43	Yes/No	Does the BMP affect this land cover classification (Mixed Forest)
51	Yes/No	Does the BMP affect this land cover classification (Shrubland)
61	Yes/No	Does the BMP affect this land cover classification (Orchards/Vineyards)
71	Yes/No	Does the BMP affect this land cover classification (Grasslands/Herbaceous)
81	Yes/No	Does the BMP affect this land cover classification (Pasture/Hay)
82	Yes/No	Does the BMP affect this land cover classification (Row Crops)
83	Yes/No	Does the BMP affect this land cover classification (Small Grains)
84	Yes/No	Does the BMP affect this land cover classification (Fallow)
85	Yes/No	Does the BMP affect this land cover classification (Urban/Recreational Grasses)
89	Yes/No	Does the BMP affect this land cover classification (Dairy)
91	Yes/No	Does the BMP affect this land cover classification (Woody Wetlands)
92	Yes/No	Does the BMP affect this land cover classification (Emergent Herbaceous Wetlands)

Table: MapSelect

Field Name	Data Type	Description
ShapeIndex	Long Integer	The index of the shape last selected in the shapefile

Table: Scenario_BMPs

Field Name	Data Type	Description
ScenarioId	Long Integer	The scenario associated with the best management practice
DrainageId	Long Integer	The drainage id associated with the best management practice
DrainageName	Text	The drainage name associated with the best management practice
BMP_Name	Text	The name of the best management practice
PercentAffected	Long Integer	The percentage of the land cover or watershed

		area the best management practice is applied to
Phos_Reduction	Long Integer	The percentage of reduction for phosphorus
N_Reduction	Long Integer	The percentage of reduction for nitrogen
NH3_NH4_N_Reduction	Long Integer	The percentage of reduction for NH_3 + NH_4 - N
NO2_N03_N_Reduction	Long Integer	The percentage of reduction for NO_2 + NO_3 - N
BOD_Reduction	Long Integer	The percentage of reduction for BOD
FecalColBac_Reduction	Long Integer	The percentage of reduction for fecal coliform bacteria
11	Yes/No	Does the BMP affect this land cover classification (OpenWater)
12	Yes/No	Does the BMP affect this land cover classification (Perennial Ice/Snow)
21	Yes/No	Does the BMP affect this land cover classification (Low Intensity Residential)
22	Yes/No	Does the BMP affect this land cover classification (High Intensity Residential)
23	Yes/No	Does the BMP affect this land cover classification (Commercial/Industrial/Transportation)
31	Yes/No	Does the BMP affect this land cover classification (Bare Rock/Sand/Clay)
32	Yes/No	Does the BMP affect this land cover classification (Quarries/Strip Mines, Gravel Pits)
33	Yes/No	Does the BMP affect this land cover classification (Transitional)
41	Yes/No	Does the BMP affect this land cover classification (Deciduous Forest)
42	Yes/No	Does the BMP affect this land cover classification (Evergreen Forest)
43	Yes/No	Does the BMP affect this land cover classification (Mixed Forest)
51	Yes/No	Does the BMP affect this land cover classification (Shrubland)
61	Yes/No	Does the BMP affect this land cover classification (Orchards/Vineyards)
71	Yes/No	Does the BMP affect this land cover classification (Grasslands/Herbaceous)
81	Yes/No	Does the BMP affect this land cover classification (Pasture/Hay)
82	Yes/No	Does the BMP affect this land cover classification (Row Crops)
83	Yes/No	Does the BMP affect this land cover classification (Small Grains)

84	Yes/No	Does the BMP affect this land cover classification (Fallow)
85	Yes/No	Does the BMP affect this land cover classification (Urban/Recreational Grasses)
89	Yes/No	Does the BMP affect this land cover classification (Dairy)
91	Yes/No	Does the BMP affect this land cover classification (Woody Wetlands)
92	Yes/No	Does the BMP affect this land cover classification (Emergent Herbaceous Wetlands)

2. GIS Data Needs

The Best Management Practices Tool requires a drainages shapefile to manage BMPs on a per drainage basis. The default installation location for this drainage shapefile is:

C:\Program Files\WRIA-1_DSS
\DSS_Data\GIS_Data\Shapefiles\Watershed\bsnwria1_v7.shp

3. Dependencies

The Best Management Practices plug-in requires the following software to be installed:

Software:
MapWindow 3.1
DSS Model Manager

Integrated Development Environment (IDE):
Visual Studio .NET 2003 Complete Install

4. Setup

The WRIA-1_DSS Installation installs the Best Management Practices Tool in the MapWindow Plugins folder.
(This is usually located at C:\Program Files\MapWindow\Plugins\ModelManager\Elements").

5. Building

To compile the Best Management Practices Tool, add the files below to a Microsoft Visual Studio .NET 2003 Visual Basic .NET Library Project.

mwBestManagementPractices project files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
BestManagementPractice.vb	Implements MapWindow Plugin and the DSS Interface routines, allowing the Best Management

	Practices Tool to act as a MapWindow Plugin and a DSS element.
ChangeDatabase.ico	An icon used for the Change Database button on the toolbar.
DBClient.vb	Contains the routines used to connect to a database and store the best management practices created.
Delete.ico	An icon used for the Delete BMP button on the toolbar.
Edit.ico	An icon used for the Edit BMP button on the toolbar.
frmConfiguration.vb	Contains the routines used to select a best management practices database.
frmCreateBMP.vb	Allows the user to create best management practices for specified drainages.
frmEditDefaultParameters.vb	Allows the user to edit default bmp parameters in the database (THIS HAS BEEN DISABLED IN THE CODE TO AVOID DATA CORRUPTION)
frmOpenScenarioBMPs.vb	Allows the user to select a previous set of BMPs for a specified scenario number.
frmSelectLayer.vb	Contains the routines used to select a layer from MapWindow's view.
New.ico	An icon used for the New BMP button on the toolbar.
Open.ico	An icon used for Open BMP button on the toolbar.
Save.ico	An icon used for the Save BMP button on the toolbar.

Add the following references to the project:

- DssIntfcLib.dll
- MapWinGIS.ocx
- MapWinInterfaces.dll
- stdole.dll
- System.dll
- System.Data.dll
- System.Drawing.dll
- System.Windows.Forms.dll
- System.Xml.dll

You are now ready to compile the project by clicking the Build Solution menu option under the menu Build.

The Best Management Practices Tool is installed in all versions of the WRIA-1 DSS installation.

Technical Documentation: Nooksack DBMS/LaunchPad

Last Revision: 3/22/04

Table of Contents

1. *Table Designs*
   - *Schema, Keys, Indexes and Structure*
   - *Table Descriptions and Notes*

2. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

3. *Setup*
   - *Setup and Configuration*
   - *Interactions between Client and Server*

4. *Building*
   - *Code Compiling*
   - *Individual Projects of the DBMS*
   - *Installations*

1. Table Designs

Database: LaunchPad

Table: Binaries
Description: Contains information about the products that the DBMS supports updating. Any product in this list will be checked to see if the local computer has an up-to-date copy, as well as giving users the option to install any missing component.

Field	Type	Size	Indexed	Primary Key	Comments
productName	Text	50	Yes	No	Name of product or component.
currentVersion	Text	50	No	No	Most recent version (version the server is distributing)
updateDate	Date/Time		No	No	Last date updated by the Update Tool
allowDownload	Boolean		No	No	True/False – allow user to download the component? [Not Implemeted]
recordnumber	Numeric		Yes	Yes	Internal record number.
lastLocalPath	Text	255	No	No	The last place the component was located on the local computer when being updated. Used for convenience from the Updater tool.

Table: BinariesFiles
Description: Contains information about the files associated with each of the products in the Binaries table.

Field	Type	Size	Indexed	Primary Key	Comments
binaryRecNumber	Numeric		Yes	No	Recordnum from the Binaries table that this record is associated with.
fullPathOnRemote	Text	400	No	No	Where this file belongs on the client computers.
filerecordnumber	Numeric		Yes	Yes	Internal record number.
VersionFileForProduct	Boolean		No	No	True/False – indicates that this file is the file which determines the version of the whole product.
downloadPath	Text	400	No	No	Location (URL) where the file may be downloaded from.
filesize	Numeric		No	No	Size of the file, in bytes.

Database: ChangeLog

Table: Changes
Description: Contains a log of who has changed data in the database. This table is filled by triggers on every SQL server table.

Field	Type	Size	Indexed	Primary Key	Comments
tablename	Text	150	No	No	The name of the table modified.
datemodified	Date/Time		No	No	Date the changes were made.
hostname	Text	150	No	No	The hostname of the computer making the changes
username	Text	150	No	No	The username (if retrievable) of the user making the changes.
appname	Text	150	No	No	Name of application making change
actiontaken	Text	150	No	No	Action taken – insert/delete/update
recordnumIFA	Numeric		No	No	Record number altered in the table, if available.

Table: CommonUsers
Description: Known users who make changes to the database. Related to the table above by username. Provides descriptive information about known users who make changes.

Field	Type	Size	Indexed	Primary Key	Comments
username	Text	255	No	No	Username corresponding to the username in Changes
commonname	Text	255	No	No	The full name or descriptive information.

2. Dependencies

The DBMS requires the following software components and modules to be installed:

Software:
- Visual Studio .NET 2002
- Microsoft SQL Server 2000 (optional)
  --This will provide administration tools which make management of the databases much easier.--
- InstallShield Express 3.5 with Service Pack 4 (or better)

Components:
- Microsoft Common Controls 6.0
- Microsoft Common Controls-2 6.0

Microsoft Common Controls-3 6.0
Microsoft FlexGrid Control 6.0
Fish Periodicity Lifestage Plotter Control (USU)
Microsoft SQL-DMO Data Object

Other parts of the DBMS/LaunchPad Suite: (described in section 4)
The DBMS/LaunchPad itself
DBMS Server *(Windows Service)*
DBMS Server Configuration Editor
DBMS Update Assistant
LaunchPad Product Updater

3. Setup

DBMS Server

The DBMS Server is a windows service. This needs to first be configured with the DBMS Server configuration tool. First, set the root path for GIS data; this is the directory containing the GIS data for which updates need to be propagated. Also set the administrator's e-mail address and a mail server which may be used to send mail, and set the interval between data reindexes. This interval should be often enough to capture data changes, but not more than every half hour.

When complete, save this configuration file to a location you'll remember.

Next, install the DBMS Server service. This may be done by copying the binary executable to a location on the server (any location is fine; typically, this is in a subdirectory of c:\Program Files). Once the binary has been copied, use the InstallUtil.exe file that should be with the binary executable. Type the following command:
InstallUtil DBMSService.exe
The command will provide a great deal of output, hopefully finishing with an "Install Successful". Now, the service is listed under Services (Start-Settings-Control Panel-Administrative Tools-Services).

Move the configuration file you saved above so that it is located next to the DBMSService.exe file. Do this before starting the service!

Now, it's necessary to decide where to place binary update components and binary installation packages. This can be anywhere you prefer. Copy these files into place, and write down the path to each of these locations.

You'll also need to create a database backup directory, where the DBMS Client will ask the server to create a database backup to download (for updating databases). Write down this path as well.

Finally, you'll need a web server such as IIS (Internet Information Services) or Apache (a free web server) installed. Configure your web server such that the directories above are accessible.

The directories which need to be accessible through the web browser, in summary:

1. GIS Data directory. This is the same directory that you configured in the DBMS Server Configuration Tool.
2. The binary component update directory, storing the binary components which will be updated with the DBMS.
3. The binary installation directory, storing the installation programs which will be downloaded by the DBMS.
4. The database backup directory. Nothing needs to be placed in this directory immediately; backups will be generated and placed here for the database updater to download.

Items number two and three may optionally be the same directory, but number 1 should not be shared with any other data items or files. It's recommended that you use a separate directory for each of these items.

Items 1, 3, and 4 are configured in the DBMS Client's configuration file; item number 2 is configured on a per-component basis in the LaunchPad database.

Write down the URLs to get to each of these directories through the web server, as they will be needed in the DBMS Client (LaunchPad) setup instructions. Also write down the actual path to the database backup directory, relative to the server's directory structure.

Next, MS SQL Server or MSDE will need to be installed on the server. Follow the installation instructions for whichever of these two you'd like to use, then attach the LaunchPad database and any other databases that you wish to update using the DBMS. There should be five user databases total: ChangeLog, LaunchPad, WRIAReportData, FlowData, and WaterQuality.

After attaching the LaunchPad database, check the contents of the Binaries table to ensure that the products that you wish to be automatically updated are listed. You may do this with the LaunchPadProductUpdater tool, or you may use a third-party database modification tool. If you choose the latter, you'll need to know this: for each Binaries recordnumber, there are one or more BinariesFiles records that are associated with it (Binaries.recordnumber -> BinariesFiles.binaryRecNumber). Also ensure that the downloadpath and fullpathonremote fields in the BinariesFiles records are correct.

At this point, the server should be configured and ready for use.

DBMS Client (LaunchPad)

The DBMS Client uses a configuration file similar to that used by the server. There is not a standalone configuration program for the DBMS Client, however. The configuration

file, DBMS.conf, may be edited by hand using Notepad or it may be written by making the appropriate changes using the Administrative Tools button in the DBMS Client itself.

Once this configuration file has been properly generated and set up, it may be distributed to all clients along with the DBMS executable. You may also include the DBMS.conf file with the binaries in an installer and distribute the installer to the clients. (A DBMS standalone installer already exists.)

The configuration file contains connection information to reach the SQL server, as well as the URLs for each of the directories listed under DBMS Server configuration. The contents of the configuration file follow in the table below. Once each of these values is configured, the DBMS Client configuration is complete.

Tag in Configuration File	Purpose	Default Value
SQLServerAddr	SQL Server address.	Nooksack.uwrl.usu.edu
SQLPort	Port number of SQL server	22
SQLUID	SQL Server Username	WRIAUser
SQLPWD	SQL Server Password	quebec41
SQLTrusted	Indicates whether to use Integrated Security.	False
UpdateDLPath	Where to download GIS data updates.	http://nooksack.uwrl.usu.edu/DBMSSource/
BinaryDLPath	Where to download binary installation packages. (Individual components configured in database.)	http://nooksack.uwrl.usu.edu/download/
DatabaseDLPath	Where to download database backups for updating databases.	http://nooksack.uwrl.usu.edu/dbupdates/
DatabaseBKPath	Path on server (relative to server) where database backups are created and stored.	e:\nookweb\dbupdates\
BinaryUpdatesPath	Location of the binary updates (NOT the installers!) on the server. Relative to the	e:\nookweb\DBMSSource\

	server.	
DataUpdatesPath	Path to GIS data updates, relative to server.	E:\nookweb\DBMSSource
ProxyUID	Username to access the proxy server with. Encrypted.	(blank)
ProxyPWD	Password to access the proxy server with. Encrypted.	(blank)

The DBMS needs to know the locations relative to the server so that it may prune directories when necessary; it also needs to know the URLs to get the files in the first place.

4. Building

Compiling the DBMS and any of the DBMS-related tools is a straightforward task. After ensuring that all of the required components discussed in item 3 above are present, load the project into Visual Studio and click the Build icon, or select Build from the menu. These instructions hold true for any of the individual projects of the DBMS.

The individual projects, their purpose, and their contents are described below.

DBMS Server (*Windows Service*)

This tool is used to maintain an index file of data available for the client tool (the LaunchPad) to download. It runs as a windows service and scans the target directory (and all subdirectories) to create the index file, taking the time between index scans and the location for all files from the configuration file next to the executable. The server also listens for requests to update the replication databases (the database backups that are zipped in the download directory). The client can send the request, then disconnect and let the server finish without an active connection; or, the client can send a request and wait for completion, receiving progress updates along the way.

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
DBMSService.vb	The actual service routines. Contains an implementation of a windows service.
ProjectInstaller.vb	This class causes the service to be installed as a *service* instead of just as a binary application.
UserConnection.vb	Contains a user connection class, a new instance of which is created every time a new user connects to the service. This is currently unused – see the DBMS source code.

DBMS Server Configuration Editor

This tool allows the configuration file used by the DBMS Server to be edited. This provides a simple, easy-to-use interface to change values such as the time between reindexes, location of GIS data, and administrator e-mail address.

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
frmConfigOptions.vb	Form containing the configuration file editor, providing a user interface.

DBMS Update Assistant

This is a tool used by the DBMS/LaunchPad when it needs to download a product update for itself. When started, it ends all instances of the DBMS it finds running, then downloads the updated DBMS/LaunchPad executable from the location specified in the LaunchPad database. After finishing, it restarts the DBMS.

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
DBMSAssist.vb	The form contains a progress meter showing activity – on load, the form performs its actions described above.

LaunchPad Product Updater

This is a tool to facilitate uploading new versions of components by programmers. The program lists all of the components available for updating and their associated files; a product may be updated, or a new product may be added. The tool uploads the files to the server and adds appropriate entries into the LaunchPad database.

Note: The "Allow User Download" feature is not implemented, i.e. all components may be downloaded by the user.

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
Form1.vb	The form contains the fields to display all product information, and allows the user to change the data or upload a new project to the server.

DBMS/LaunchPad Main Project

This is the main DBMS/LaunchPad. When the LaunchPad is first run, it will scan the computer for all requisite components and data items. If something is missing, it will provide the user with the opportunity to download and install it. If all necessary components are present, the DBMS will allow the user to edit data on the master server (if the Administrative Edition is in use), or the user may launch the Nooksack DSS. The user may also check for updates – this causes the DBMS to scan the local computer to see if any data items or binary components on the remote computer are newer than the local components/data items. If newer components are found, the user may download and install the updates. The LaunchPad may also be used to send a request to the server to update the databases that are available for download ("replication databases").

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
frmChangeLogView.vb	This form shows the log of changes made to the database.
frmDlStatus.vb	This is a simple status indicator used to show the progress in updating the remote databases. This is used only if the user decides to wait for the server to finish, rather than disconnecting.
frmEditFishDist.vb	Allows the user to change fish distribution for a drainage.
frmEditFishHabitatRestoration.vb	Allows the user to change the text block representing fish habitat restoration projects.
frmEditGlossary.vb	Allows editing of the glossary terms and definitions.
frmEditISF.vb	Allows editing of instream flow requirements.
frmEditOverview.vb	Allows editing of the report overview. The disclaimer is static and may not be edited.
frmEditPeriodicity.vb	Lets the user edit fish periodicity for a given dataset, drainage, and stream segment.
frmEditRecreation.vb	Allows editing of recreational data – location and activities – for a drainage.
frmEditStreamClosures.vb	Allows editing of stream closure data for a drainage.
frmEditWashingtonCensus.vb	Allows editing of the census data used in the report.
frmEditZoningAbbrev.vb	Lets the user edit zoning abbreviations; i.e. R-Forest = Rural Forest.
frmFlowData.vb	Lists flow data collected for a given station. Allows user to select the station from a combo box.
frmFlowDataAgenciesImport.vb	Allows flow data agencies to be imported into the database from a delimited text file.

frmFlowDataCommentsImport.vb	Allows flow data comments to be imported into the database from a delimited text file.
frmFlowDataGageStationImport.vb	Allows flow data gage stations to be imported into the database from a delimited text file.
frmFlowDataImport.vb	Allows flow data to be imported into the database from a delimited text file.
frmGridEditor.vb	Provides a generic form and datagrid to edit large tables. The specific database and table are set at runtime, as are the window caption, window labels, and other details. The table can then be edited through the bound datagrid. These settings are accomplished through public data members (.settings_*)
frmImportPreview.vb	Shows a preview of what the data will look like once imported from the text file, to ensure it's as desired before performing an actual import. This is a generic screen used from all of the import screens.
frmProxyConfiguration.vb	Web proxy configuration screen; appears whenever a proxy server that requires authentication is detected.
frmREditAddAggrDrainage.vb	Inserts a drainage into a predefined aggregation.
frmREditAggregations.vb	Allows the user to add and remove drainages from predefined aggregations, or add and remove predefined aggregations.
frmREditDrainages.vb	Allows editing of the drainage information, e.g. name, description, area.
frmRemoteEdit.vb	This is the "launcher screen" for remote data editors; after a user has selected which database to edit, gives the user a list of tables they may edit. Clicking a table name launches the appropriate data editor form. Also allows the user to alter database connection information and alter installation and data download URLs.
frmSectionEdit.vb	Allows the user to edit data sources, headings, commentary, and visibility for all of the sections within the report. These changes are made on the master server, as with everything in DBMS, for propagation to clients.
frmShellfishByDrainage.vb	Allows editing of the text block representing shellfish comments for each drainage.
ReportMain.vb	This contains all of the preparation the reports themselves. Maps are generated in this module as well.
frmSplash.vb	This is the main screen – This does the component checking to ensure requisite components are present. This form is also responsible for the data update scanning.

frmWaterQuality.vb	Allows the user to edit Water Quality data, including import water quality data (Calls importer screens below).
frmWellsByDrainage.vb	Allows altering the number of wells per drainage as well as changing the well-related comments for the drainage.
frmWQAgencyCodeImport.vb	Allows the user to import water quality agency code data from a delimited text file.
frmWQCommentCodeImport.vb	Allows the user to import water quality comment code data from a delimited text file.
frmWQDataCodeImport.vb	Allows the user to import water quality data from a delimited text file.
frmWQDataImport.vb	Allows the user to import water quality data from a delimited text file. All code descriptions must be imported separately.
frmWQParameterCodeImport.vb	Allows the user to import water quality parameter code data from a delimited text file.
frmWQQAQCCodeImport.vb	Allows the user to import water quality QAQC (quality assurance/quality check) codes from a delimited text file.
frmWQSourceDatabaseImport.vb	Allows the user to import water quality source database codes from a delimited text file.
frmWQStationImport.vb	Allows the user to import water quality stations from a delimited text file.
web.avi	This is the file transfer animation shown during a file download or data update.

Installers

An installer exists for only the DBMS/LaunchPad, as it is the only project intended to be widely distributed. The Update Assistant is automatically downloaded when needed by the DBMS, so no installer is necessary for it.

The DBMS/LaunchPad tool will download other components or installers as necessary, so no other installations need to be built and shipped with this one.

Technical Documentation: The Habitat Time Series Model

Last Revision: 06/21/06

Table of Contents

1. *Input Habitat Database Table Design*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

2. *Input Daily Flow Database Table Design and Data Needs*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*
   - *Scenario Run Data Needs*

3. *Input Model Nodes Shapefile Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*
   - *Necessary Fields*
   - *Selecting Nodes for Modeling*

4. *Weighted Usable Area (WUA) Data Needs*
   - *Extrapolation Method Information*

5. *Output Database Table Design*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

6. *Dependencies*
   - *Software*
   - *Modules and Components*
   - *Integrated Development Environment (IDE)*

7. *Setup*
   - *Setup*

8. *Code Compiling*
   - *Project Files*
   - *Reference Settings*

1. Input Habitat Database Table Design

Following is a list of tables that must be included with the Input Habitat Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters.

Table: tblLifestages
Description: Contains the Lifestage Information for each of the Species that are available.

Field	Type	Size	Key Field	Primary Key	Comments
Lifestage_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	Unique ID for each Lifestage available. It corresponds with the *Lifestage_ID* Field in the *tblPeriodicity* table, the *tblWUA_MethodB* table, and the *tblWUA_MethodC* table.
Species_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Species that this Lifestage belongs to. It corresponds with the *Species_ID* Field in the *tblSpecies* table.
Lifestage_Name	Text	255	No	No	Name of the Lifestage.

Table: tblNodes
Description: Contains the Node Info (ID, Drainage Info, Reach Info) for all available nodes.

Field	Type	Size	Key Field	Primary Key	Comments
Pk_NodeID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Node.
Node_ID	Number	Long Integer	Yes (No Duplicates)	No	This is the ID for this Node. It is the value that corresponds with NodeID that is stored in the Input Model Nodes Point Shapefile. It also corresponds with the *Node_ID* Field in the *tblPeriodicity* table, the *tblWUA_MethodB* table, and the *tblWUA_MethodC*.

Drainage_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Drainage that this node belongs to.
Drainage_Name	Text	255	No	No	This is the Name of the Drainage that this node belongs to.
Reach_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Reach that this node belongs to.
Reach_Name	Text	255	No	No	This is the Name of the Reach that this node belongs to.

Table: tblPeriodicity

Description: Contains the Bi-Monthly Periodicity values.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
Periodicity_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of Periodicity values in the table.
Node_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Node that this set of Periodicity values belong to. It corresponds with the *Node_ID* Field in the *tblNodes* table.
Lifestage_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Fish that this set of Periodicity values belong to. It corresponds with the *Lifestage_ID* Field in the *tblLifestages* table.
Jan_1	Number	Integer	No	No	Periodicity Value for the 1st half of January for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jan_2	Number	Integer	No	No	Periodicity Value for the 2nd half of January for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Feb_1	Number	Integer	No	No	Periodicity Value for the 1st half of February for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Feb_2	Number	Integer	No	No	Periodicity Value for the 2nd half of February for the given fish, node

					pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Mar_1	Number	Integer	No	No	Periodicity Value for the 1st half of March for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Mar_2	Number	Integer	No	No	Periodicity Value for the 2nd half of March for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Apr_1	Number	Integer	No	No	Periodicity Value for the 1st half of April for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Apr_2	Number	Integer	No	No	Periodicity Value for the 2nd half of April for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
May_1	Number	Integer	No	No	Periodicity Value for the 1st half of May for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
May_2	Number	Integer	No	No	Periodicity Value for the 2nd half of May for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jun_1	Number	Integer	No	No	Periodicity Value for the 1st half of June for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jun_2	Number	Integer	No	No	Periodicity Value for the 2nd half of June for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jul_1	Number	Integer	No	No	Periodicity Value for the 1st half of July for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jul_2	Number	Integer	No	No	Periodicity Value for the 2nd half of July for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*

Aug_1	Number	Integer	No	No	Periodicity Value for the 1st half of August for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Aug_2	Number	Integer	No	No	Periodicity Value for the 2nd half of August for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Sep_1	Number	Integer	No	No	Periodicity Value for the 1st half of September for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Sep_2	Number	Integer	No	No	Periodicity Value for the 2nd half of September for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Oct_1	Number	Integer	No	No	Periodicity Value for the 1st half of October for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Oct_2	Number	Integer	No	No	Periodicity Value for the 2nd half of October for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Nov_1	Number	Integer	No	No	Periodicity Value for the 1st half of November for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Nov_2	Number	Integer	No	No	Periodicity Value for the 2nd half of November for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Dec_1	Number	Integer	No	No	Periodicity Value for the 1st half of December for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Dec_2	Number	Integer	No	No	Periodicity Value for the 2nd half of December for the given fish, node

					pair. *Note: 0=Not Present, 1=Present, 2=Peak*

Table: tblSpecies
Description: Contains the Species Information (ID, Name) for each one available.

Field	Type	Size	Key Field	Primary Key	Comments
Species_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Species. It corresponds with the Species_ID Field in the tblLifestages table.
Species_Name	Text	255	No	No	This is the Name of the Species.

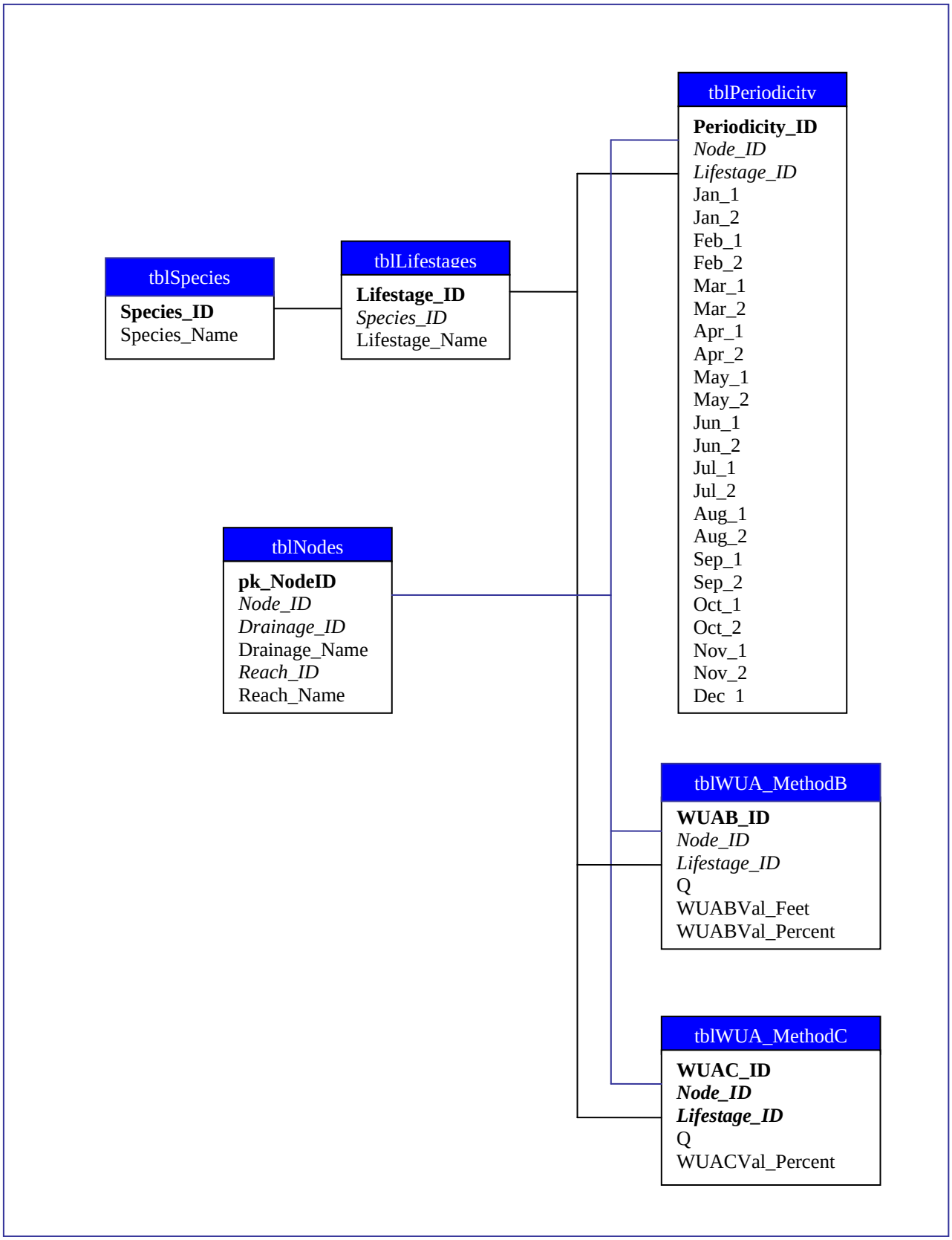
Table: tblWUA_MethodB
Description: Contains the Weighted Usable Area (WUA) data for the Method B Data Extrapolation Method.

Field	Type	Size	Key Field	Primary Key	Comments
WUAB_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	A unique ID for each Weighted Usable Area (WUA) value.
Node_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Node that this Weighted Usable Area (WUA) value belongs to. It corresponds with the *Node_ID* Field in the *tblNodes* table.
Lifestage_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the fish that this Weighted Usable Area (WUA) value belongs to. It corresponds with the *Lifestage_ID* Field in the *tblLifestages* table.
Q	Number	Double (Fixed: 3 decimal places)	No	No	The Flow value for this Weighted Usable Area (WUA) value.
WUABVal_Feet	Number	Double (Fixed: 3 decimal	No	No	The Weighted Usable Area (WUA) value where Units = Ft/1000 Ft.

		places)			
WUABVal_Percent	Number	Double (Fixed: 3 decimal places)	No	No	The Weighted Usable Area (WUA) value where Units = Percent of Maximum.

Table: tblWUA_MethodC
Description: Contains the Weighted Usable Area (WUA) data for the Method C Data Extrapolation Method.

Field	Type	Size	Key Field	Primary Key	Comments
WUAC_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	A unique ID for each Weighted Usable Area (WUA) value.
Node_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Node that this Weighted Usable Area (WUA) value belongs to. It corresponds with the *Node_ID* Field in the *tblNodes* table.
Lifestage_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the fish that this Weighted Usable Area (WUA) value belongs to. It corresponds with the *Lifestage_ID* Field in the *tblLifestages* table.
Q	Number	Double (Fixed: 3 decimal places)	No	No	The Flow value for this Weighted Usable Area (WUA) value.
WUACVal_Percent	Number	Double (Fixed: 3 decimal places)	No	No	The Weighted Usable Area (WUA) value where Units = Percent of Maximum


tblPeriodicitv
Periodicity_ID
Node_ID
Lifestage_ID
Jan_1
Jan_2
Feb_1
Feb_2
Mar_1
Mar_2
Apr_1
Apr_2
May_1
May_2
Jun_1
Jun_2
Jul_1
Jul_2
Aug_1
Aug_2
Sep_1
Sep_2
Oct_1
Oct_2
Nov_1
Nov_2
Dec 1
tblSpecies
Species_ID
Species_Name
tblLifestages
Lifestage_ID
Species_ID
Lifestage_Name
tblNodes
pk_NodeID
Node_ID
Drainage_ID
Drainage_Name
Reach_ID
Reach_Name
tblWUA_MethodB
WUAB_ID
Node_ID
Lifestage_ID
Q
WUABVal_Feet
WUABVal_Percent
tblWUA_MethodC
WUAC_ID
Node_ID
Lifestage_ID
Q
WUACVal_Percent

1782

2. Input Daily Flow Database Table Design and Data Needs

Following is a list of tables that must be included with the Input Daily Flow Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters.

Table: Parameter_Code Descriptions
Description: Contains the Code and Name for the available Parameters.
NOTE: *The Parameters correspond with the Scenario Runs from the Water Quantity Model.*

Field	Type	Size	Key Field	Primary Key	Comments
Parameter_Code	Number	Long Integer	Yes (Duplicates OK)	No	This is a unique ID or Code for each Parameter. It corresponds with the *Parameter_Code* Field in the *WQData* table.
Parameter_Name	Text	50	No	No	This is the Name of the Parameter. *For example: ScenarioID_StreamFlow.*

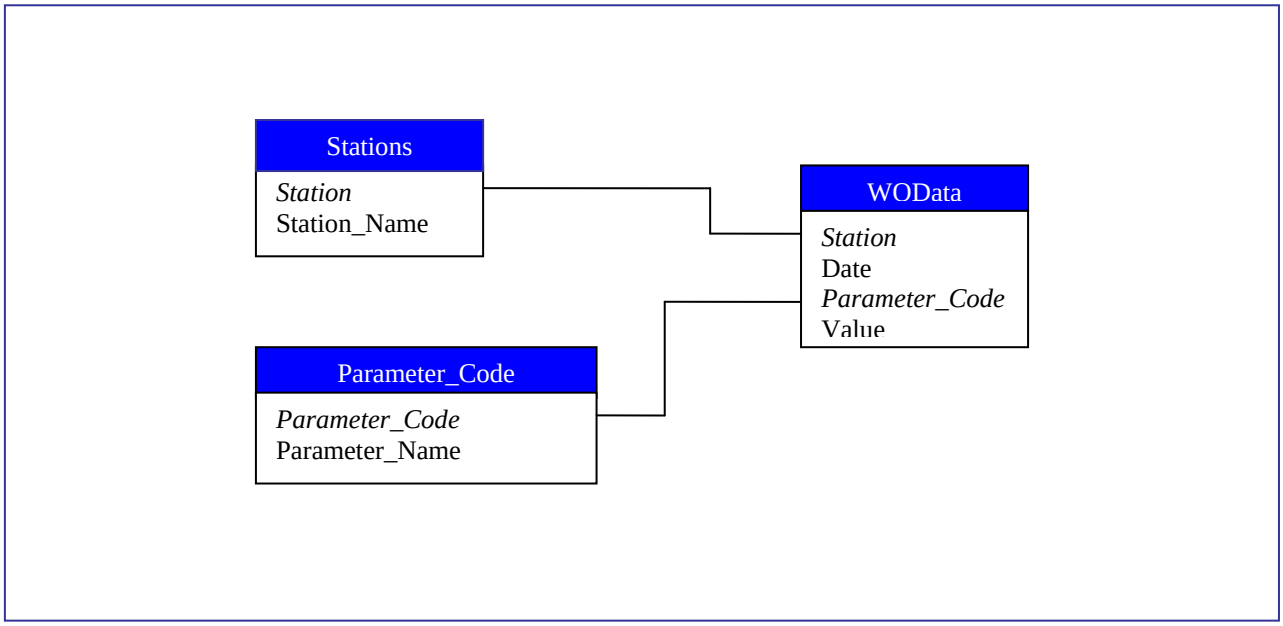
Table: Stations
Description: Contains the ID and Name for the available Stations.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	50	Yes (Duplicates OK)	No	This is a unique ID for each Station. It corresponds with the *Station* Field in the *WQData* table.
Station_Name	Text	50	No	No	This is the Name for the Station.

Table: WQData
Description: Contains the modeled Daily Stream Flow data from the Water Quantity Model.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	50	Yes (Duplicates OK)	No	This is the Station that this value belongs to. It corresponds with the *Station* Field in the *Stations* table.
Date	Date/Time		No	No	This the Date/Time of this value.

Parameter_ Code	Number	Long Integer	Yes (Duplicates OK)	No	This is the Parameter Code for this sample. It corresponds with the *Parameter_Code* Field in the *Parameter_Code Descriptions* table.
Value	Number	Double	No	No	This is the Daily Streamflow value.



Input Scenario Run Data Needs

The daily flow values written by the Water Quantity Model for a scenario run are all stored with the same Parameter Name. These Parameters follow the naming convention: *ScenarioID*_StreamFlow, where the *ScenarioID* is the Scenario ID from the model run in the DSS Scenario Builder. Although the Habitat Time Series Model with most likely be run in conjunction with the Water Quantity Model and the selected Input Scenario Run, the Parameter Names in *Parameter_Code Descriptions* table in the Input Daily Flow Database, will follow the described naming convention, it does not have to be selected from a Water Quantity Model Run, so it can be named anything.

3. Input Model Nodes Shapefile Data Needs

There are two types of data required for the Input Model Nodes Shapefile: a point shapefile path, and a *Node ID* shapefile field.

- Point Shapefile – this shapefile needs to be in the same project as the other shapefiles in your project. It contains the point locations of each of the Model Nodes. Connection Name – this is the name of the Time Series Analyst Connection.

- *Node ID* Shapefile Field – this field is a String field that contains the Node ID for each of the Model Nodes. It corresponds with the *Node_ID* Field in the *tblNodes* table for the Input Habitat Database, and with the *Station* Field in the *Stations* table for the Input Daily Flow Database.
  NOTE: If the Node ID in the shapefile is not found in either of the Input Databases, then the model will still run, values will just not be computed for those nodes, and there will be an entry in the log explaining why.

In order to specify the Nodes to model, the Model Nodes Shapefile must be selected and added to the project. Then you can either select the which Nodes to model by specifying the Node ID's or by going to map and selecting with the Map Window Selection tools.

4. Weighted Usable Area (WUA) Data Needs

To be able to compute and write results to the Output Database for the Habitat Time Series Model, the Weighted Usable Area (WUA) Data must be present in the Input Habitat Database (see *Section 1: Input Habitat Database Table Design*). If it is not present, then no data will be written to the Output Database, a message will be in the log explaining why, and the Habitat Time Series Model will complete successfully.

The Weighted Usable Area (WUA) data is located in two tables: *tblWUA_MethodB* and *tblWUA_MethodC*. The table *tblWUA_MethodB* is used to store the data that was created using the Method B Data Extrapolation Method. The table *tblWUA_MethodC* is used to store data that was created using the Method C Data Extrapolation Method. The data in these tables is stored by reference to the Node, Fish, and Flow for that Weighted Usable Area (WUA) value.

5. Output Database Table Design

Following is a list of tables that must be included with the Output Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters.

Table: Fish
Description: Contains the Fish (Species and Lifestage) that have been modeled with the Habitat Times Series Model.

Field	Type	Size	Key Field	Primary Key	Comments
Fish_ID	Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Fish. A Fish is a Species and Lifestage pair. This values is passed from the Habitat Time Series Model. It corresponds

					with the *Lifestage_ID* Field in the *tblPeriodicity* table for the Input Habitat Database (see S*ection 1: Input Habitat Database Table Design*).
Species_Name	Text	50	No	No	This is the name of the Species that this Fish belongs to.
Lifestage_Name	Text	50	No	No	This is the name of the Lifestage that this Fish belongs to.

Table: Notes
Description: Contains any Notes about the Model Nodes (Stations).

Field	Type	Size	Key Field	Primary Key	Comments
NoteID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Note.
Station	Text	50	Yes (Duplicates OK)	No	This is the Node (Station) that the Note belongs to. It corresponds with the *Node_ID* Field in the *tblNodes* table for the Input Habitat Database (see *Section 1: Input Habitat Database Table Design*) and the *Station* Field in the *Stations* table for the Input Daily Flow Database (see *Section 2: Input Daily Flow Database Design and Data Needs*).
Notes	Memo		No	No	The Notes about the given Node (Station).

Table: Parameter_Code Descriptions
Description: Contains the Code and Name for the available Parameters.
NOTE: The parameters are created by the model using the Daily Flow Scenario Run and the Weighted Usable Area (WUA) Units.

Field	Type	Size	Key Field	Primary Key	Comments
Parameter_Code	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Parameter.
Parameter_Name	Text	100	No	No	This is the Name of the Parameter. Its value = *"Habitat Times Series Scenario Run ID"*_HabTS

					: *Daily Flow Scenario Run"* (*"WUA Units"*).

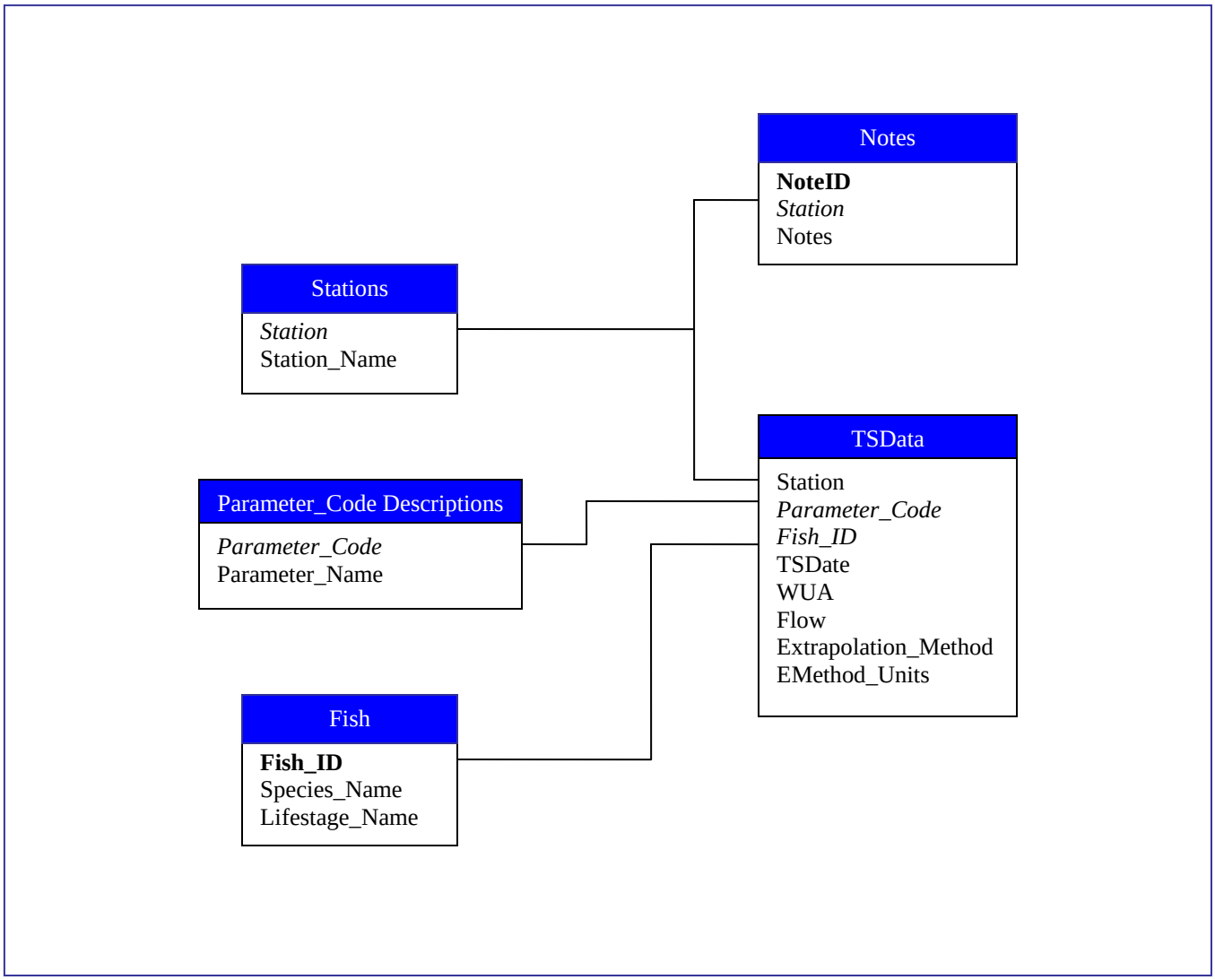
Table: Stations
Description: Contains the ID and Name for the available Stations (Nodes).

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	50	Yes (No Duplicates)	Yes	This is a unique ID for each Station (Node). It is the NodeID for the modeled nodes in the Habitat Time Series Model. It corresponds with the *Node_ID* Field in the *tblNodes* table for the Input Habitat Database (see *Section 1: Input Habitat Database Table Design*) and the *Station* Field in the *WQData* table for the Input Daily Flow Database (see *Section 2: Input Daily Flow Database Table Design and Data Needs*).
Station_Name	Text	50	No	No	This is the description for the Station. *NOTE: It is often the same value as the* Station *Field.*

Table: TSData
Description: Contains the modeled Habitat Time Series data.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	255	No	No	This is the Station that this modeled value belongs to. It corresponds with the *Station* Field in the *Stations* table.
Parameter_Code	Number	Long Integer	Yes (Duplicates OK)	No	This is the Parameter Code that this modeled value belongs to. It corresponds with the *Parameter_Code* Field in the *Parameter_Code Descriptions* table.
Fish_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the Fish that this modeled value belongs to. It corresponds with the *Fish_ID* Field in the *Fish* table.
TSDate	Date/Time		No	No	This is the Date/Time for this modeled value.
WUA	Number	Double	No	No	This is the modeled Weighted

					Usable Area (WUA) value.
Flow	Number	Double	No	No	This is the Flow for this modeled value.
Extrapolation_Method	Text	50	No	No	This is the Weighted Usable Area Data Extrapolation Method used in the Input Weighted Usable Area (WUA) Data for this modeled value. *NOTE: Valid values are: Method B or Method C*
EMethod_Units	Text	50	No	No	This is the units of the Weighted Usable Area Data Extrapolation Method used in the Input Weighted Usable Area (WUA) Data for this modeled value. *NOTE: Valid values are: Ft^2/1000Ft or Percent of Maximum Habitat (%)*



6. Dependencies

The Time Series Analyst plug-in requires the following software components and modules to be installed:

Software:
MapWindow 3.1
Visual Studio .NET 2003

Modules and Components:
DSS Interface Definitions
MapWindow Interfaces
MapWinGIS Components

Integrated Development Environment (IDE)
Visual Studio .NET 2003

7. Setup

To begin using the Habitat Time Series Model, you must first load the Scenario Builder/DSS Plug-in into Map Window. Once it is loaded, click on the DSS Menu located on the Map Window Menu bar and select either Scenario Builder, or the name of the Scenario that you want to run. Once the Scenario Builder is and the Habitat Time Series Model is added to the current Scenario and the Run Date values selected, Run the Scenario. If all of the input and output parameters have been correctly set and they are all accessible, then the Habitat Time Series Model will run for the selected data. After it is finished the user may view the run log if they desire. If any errors occur during the run, an error log will appear explaining the error.

If the input and output data associations have not been properly set, then double click on the Habitat Time Series scenario node, and a Parameters Form will appear allowing you to edit the input and output data. Then, after the parameters have been properly associated, Run the Scenario and after it is finished, the user may view the run log if they desire or if any errors occurred, then an error log will appear explaining the error.

See the User's Manual for more information and details on setting the parameters for the Habitat Time Series Model.

8. Code Compiling

Compiling the Time Series Analyst is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 6: Dependencies* are present, load the project into Visual Studio .Net 2003. This DSS Model was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsFlow.vb	Contains a class for accessing and storing the Date/Flow data from the Input Daily Flow Database (see *Section 2: Input Daily Flow Database Table Design and Data Needs*) for a fish.
clsMainPhabTS.vb	Contains a class that implements the Map Window plug-in interface and the DSS Model Interface and other functions for editing the parameters for and running the Habitat Time Series Model.
clsPeriodicity.vb	Contains an enumeration and a class for accessing and storing the Periodicity Data from the Input Habitat Database (see *Section 1: Input Habitat Database Table Design*).

clsWUA.vb	Contains a class for accessing and storing the calculated Flow and Weighted Usable Area (WUA) values (Output values) created when the model is run for a node.
clsWUAInput.vb	Contains a class for access and storing the Weighted Usable Area (WUA) values from the Input Habitat Database (see *Section 1: Input Habitat Table Design* and *Section 4: Weighted Usable Area (WUA) Data* Needs). It also contains a class for calculating the output WUA, Flow values.
frmConfiguration.vb, frmConfiguration.resx	A Database Configuration form. This form is shown when the user edits an Input or Output Database (see *Section 1: Input Habitat Database Table Design* or *Section 2: Input Daily Flow Database Table Design and Data Needs* or *Section 5: Output Database Table Design*). It allows the user to specify settings for an SQL Database (the Server Address, User ID and Password, and Database Name) or the Path for an Access Database. The connection for the specified values can then be tested to make sure a valid database is specified. To save the selected settings, click on the Save Changes button; to cancel the changed settings, close the form with the Red X button at the top of the form. This form will only be shown if the user decides to edit any of the Input or Output databases already associated with the Habitat Times Series model.
frmParameters.vb, frmParameters.resx	A form that allows the user to view and edit the set the Input and Output Parameters for the Habitat Time Series Model.
frmSelectLayer.vb, frmSelectLayer.resx	A form to select the Input Model Nodes point shapefile for the Habitat Time Series Model (see *Section 3: Input Model Nodes Shapefile Data Needs*). It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk. This form will only be shown if the user decides to edit the shapefile already associated with the Habitat Time Series Model.
frmSelectNodes.vb, frmSelectNodes.resx	A form that allows the user to select the nodes to model, either from the Map Window Map or from the list of available nodes (see *Section 3: Input Model Nodes Shapefile and Data Needs*). This form will only be shown if the user decides to edit the selected nodes already associated with the Habitat Time Series Model.
frmSelectWUA.vb frmSelectWUA.resx	A form that allows the user the select the Weighted Usable Area (WUA) Data Extrapolation Method and Units (see *Section 4: Weighted Usable Area (WUA) Data Needs*). This form will only be shown if the user decides to edit the Weighted Usable Area Extrapolation Data already associated with the Habitat Time Series Model.
modDatabase.vb	Contains functions for access and writing to the Input and Output database, and the variable definitions for the Input

	and Output Database types, table and field names, and the SQL default values.
modGlobals.vb	Contains the definitions for variables used throughout the project, such as MapWindow variables, model variables, parameter variables, error variables, and others.
modParameters.vb	Contains the variable definitions for the Input Parameter Names.
modShapefile.vb	Contains the variable definitions and functions for accessing data on the associated Model Nodes Shapefiles (see *Section 3: Input Model Nodes Shapefile Data Needs*).
modUtils.vb	Contains functions used throughout the project for reporting errors, accessing files, and other functionality.

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 6: Dependencies*).

Reference Settings:

ADODB	CopyLocal = False
DssIntfcLib	CopyLocal= True
MapWinGIS	CopyLocal = False
MapWinInterfaces	CopyLocal = False
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwPhabTimeSeriesModel.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwTimeSeries* subdirectory in the

\Plugins\ModelManager\Elements\mwPhabTimeSeriesModel

folder. Next time that MapWindow is run, if the mwPhabTimeSeriesModel.dll was built to the correct folder, the updated changes to the Habitat Time Series Model will be available.

Technical Documentation:
Lake Whatcom Water Quality Model

Last Revision: Feb 21, 2006

Table of Contents

1. *Table Designs*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*

2. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

3. *Setup*
   - *Setup*
   - *Customizations*

4. *Building*
   - *Code Compiling*
   - *Installations*

1. Table Designs

Land Cover Database: LandCoverSummary.mdb
(See "Technical Documentation For Land Cover Summarizer" for table designs)

Best Management Practices Database: BestManagementPractices.mdb
(See "Technical Documentation For Best Management Practices Tool" for table designs)

Lake Whatcom Water Quality Parameters Database: LakeWhatcomWQParameters.mdb

Table: EMCs

Field Name	Data Type	Description
DrainageID	Long Integer	WRIA 1 Drainage ID
Land_Cover_Code	Long Integer	The associated land cover class id
EMC_TN	Text	Expected mean concentration (EMC) for total nitrogen (mg/L)
EMC_NH3	Text	EMC for ammonia (mg/L)
EMC_N03	Text	EMC for nitrate (mg/L)
EMC_TP	Text	EMC for total phosphorus (mg/L)
EMC_BOD	Text	EMC for BOD (mg/L)
EMC_FC	Text	EMC for fecal coliform (mg/L)

Table: Input Air Temperatures

Field Name	Data Type	Description
Date	Date / Time	The date the temperature data was recorded
AvgTemperature	Double	Average daily temperature at Abbotsford Canada in degrees C
AirTempFlag	Text	Flag for average daily air temperatures from Abbotsford data

Table: Nodes

Field Name	Data Type	Description
DrainageID	Long Integer	WRIA 1 Drainage ID
NodeID	Long Integer	ID of water quantity model nodes contained within the drainage

Table: Parameters

Field Name	Data Type	Description
DrainageID	Long Integer	WRIA 1 Drainage ID
DrainageName	Text	Name of the WRIA 1 Drainage
AreaAcres	Double	Area of the drainage in acres
Areakm2	Double	Area of the drainage in square kilometers
StreamDOPercentSat	Double	Streamflow DO Percent Saturation Parameter (Calibration Parameter)
SurfaceDOPercentSat	Double	Surface Flow DO Percent Saturation Parameter (Calibration Parameter)
AirSoilSlope	Double	Slope of Air Temperature Surface Soil Temperature Relationship (Calibration Parameter)
AirSoilIntercept	Double	Intercept of Air Temperature Surface Soil Temperature Relationship (Calibration Parameter)
TStreamSlope	Double	Slope of linear relationship between air temperature and stream flow temperature (Calibration Parameter)
TStreamIntercept	Double	Intercept of linear relationship between air temperature and stream flow temperature (Calibration Parameter)
QbTemp	Double	Simulation Base Flow Temperature Deg. C
QbDO	Double	Base Flow Dissolved Oxygen Concentration (mg/L)
Cb_TN	Double	Base Flow Total Nitrogen Concentration (mg/L)
Cb_NH3	Double	Base Flow Ammonia Concentration (mg/L)
Cb_NO3	Double	Base Flow Nitrate Concentration (mg/L)
Cb_TP	Double	Base Flow Total Phosphorus Concentration (mg/L)
Cb_BOD	Double	Base Flow BOD Concentration (mg/L)
Cb_FC	Double	Base Flow Fecal Coliform Bacteria Concentration (#/100 mL)

Table: ZZZ_Land_Cover_Classes (NEW table in database)

Field Name	Data Type	Description
LC_ID	Double	
LC_Code	Double	
LC_Category	Text	
LC_Description	Text	

2. Dependencies

The Lake Whatcom Water Quality Model requires the following software to be installed:

Software:

MapWindow 3.1

DSS Model Manager

Integrated Development Environment (IDE):

Visual Studio .NET 2003 Complete Install

The Lake Whatcom Water Quality Model requires the data output by the following DSS elements:

Land Cover Summary Tool

Best Management Practices Tool

3. Setup

The WRIA-1_DSS Installation installs the Lake Whatcom Water Quality Model in the MapWindow Plugins folder.
(This is usually located at C:\Program Files\MapWindow\Plugins\ModelManager\Elements").

4. Building

To compile the Best Management Practices Tool, add the files below to a Microsoft Visual Studio .NET 2003 Visual Basic .NET Library Project.

mwLakeWhatcomWQModel project files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsLakeWhatcomWQModel.vb	Runs the Lake Whatcom water quality model given the settings provided.
DBClient.vb	Contains the routines used to connect to a database.
DirectoryPicker.vb	Custom component which allows a user to select a directory from the computer's file system.
frmConfiguration.vb	Contains the routines used to select a database.
frmInputs.vb	Allows the user to select the location of input and output data for the model.
frmSelectLayer.vb	Contains the routines used to select a layer from MapWindow's view.
frmSelectModelDirectory.vb	A form containing a DirectoryPicker component which allows the user to select a directory from the file system, or a directory to be created by the Water Quantity Model in

	the current DSS Scenario Run.
WaterQualityModel.vb	Implements MapWindow Plugin and DSS Interface routines, allowing the Lake Whatcom Model to act as a MapWindow Plugin and a DSS element.

Add the following references to the project:

- DssIntfcLib.dll
- MapWinGIS.ocx
- MapWinInterfaces.dll
- mwBestManagementPractice.dll
- System.dll
- System.Data.dll
- System.Drawing.dll
- System.Windows.Forms.dll
- System.Xml.dll

You are now ready to compile the project by clicking the Build Solution menu option under the menu Build.

Lake Whatcom Waterbody Response Model project files

File Name	Purpose
resenlwfpath.f95	Lake Whatcom Waterbody Response Model Fortran 95 source file
apifunctions.f95	Library of file management routines through Windows API
ConvertwsmodelOutputPath.f95	Conversion of LWWLM output to binary format for Fortran model use
mapi.bat `lf95 resenlwfpath apifunctions.f95 ConvertwsmodelOutputPath -exe rp.exe -g -trace -ml winapi`	Fortran compiler script for creating LWWRM executable

The rp.exe file is built using the Lahey Professional Fortran compiler version 5.6. It has no additional dependencies.

The Lake Whatcom Water Quality Model is installed in all versions of the WRIA-1 DSS installation.

Technical Documentation: Land Cover Changer Tool

Last Revision: Feb 21, 2006

Table of Contents

1. *GIS Data Needs*
   - *Location of GIS Data*
   - *Tags on Layers*
2. *Dependencies*
   - *Software*
   - *Integrated Development Environment (IDE)*
3. *Setup*
   - *Setup*
   - *Customizations*
4. *Building*
   - *Code Compiling*
   - *Installations*

1. GIS Data Needs

The Land Cover Changer Tool makes changes to land cover grids supported by MapWindow. These grids include ASCII grids (*.asc), and binary grids (*.bgd). The Land Cover Changer takes a grid as input, and outputs a grid reflecting the changes selected by the user.

The Land Cover Changer Tool sets the tag of a land cover layer in MapWindow to the string value “LANDCOVER”.

2. Dependencies

The Land Cover Changer plug-in requires the following software to be installed:

Software:
MapWindow 3.1
Visual Studio .NET 2003 Complete Install

3. Setup

The WRIA-1_DSS Installation installs the Land Cover Changer Tool in the MapWindow Plugins folder.
(This is usually located at C:\Program Files\MapWindow\Plugins\ModelManager\Elements”).

4. Building

To compile the Land Cover Changer, add the files below to a Microsoft Visual Studio .NET 2003 Visual Basic .NET Library Project.
mwChangeLandCover project files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
BULLSEYE.CUR	Bullseye cursor used for indicating the next mouse click will close the current land cover change polygon being drawn.
ChangeLandCoverPlugin.vb	Implements MapWindow Plugin and DSS Interfrace routines, allowing Land Cover Changer to act as a MapWindow Plugin and a DSS element.
Common.vb	Common functions used by the land cover changer.
CURSOR.CUR	Cursor used for selecting polygons in shapefile for land cover change boundaries.
frmChangeLandCover.vb	Contains the routines used to create land cover change boundary polygons.
frmSelectChanges.vb	Contains the routines used to select what land cover changes

	to perform inside a selected boundary.
frmSelectLayer.vb	Contains the routines used to select a layer from MapWindow's view.
frmSettings.vb	Contains the routines used to select the settings for the Land Cover Change Tool.
OpenDirectoryDialog.vb	Contains the routines used to select a directory to be used by the Land Cover Change Tool.
PENCIL.CUR	Pencil cursor used when drawing a land cover change boundary on MapWindow's view.

Add the following references to the project:

- DssIntfcLib.dll
- MapWinGIS.ocx
- MapWinInterfaces.dll
- Microsoft.VisualBasic.Compatibility.dll
- stdole.dll
- System.dll
- System.Data.dll
- System.Design.dll
- System.Drawing.dll
- System.Windows.Forms.dll
- System.Xml.dll

You are now ready to compile the project by clicking the Build Solution menu option under the menu Build.

The Land Cover Changer Tool is installed in all versions of the WRIA-1 DSS installation.

Technical Documentation: Land Cover Summarizer Tool

Last Revision: Feb 22, 2006

Table of Contents

1. *Table Designs*
   - *Schema, Keys, Indexes, and Structure*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*

2. *GIS Data Needs*
   - *Location of GIS Data*

3. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*
   - *Customizations*

5. *Building*
   - *Code Compiling*
   - *Installations*

1. Table Designs

Database: LandCoverSummary

Field Name	Data Type
ScenarioID	Number
WatershedID	Number
WatershedName	Text
LandCoverValue	Number
Area	Number

2. GIS Data Needs

The Land Cover Summarizer Tool summarizes the land cover types within each shape in a watershed shapefile using a land cover grid supported by MapWindow. These grids include ASCII grids (*.asc), and binary grids (*.bgd). The Land Cover Summarizer takes a shapefile and a grid as input, and outputs a list of land cover summaries for each shape in the shapefile.

3. Dependencies

The Land Cover Summarizer plug-in requires the following software to be installed:

Software:

MapWindow 3.1

Visual Studio .NET 2003 Complete Install

4. Setup

The WRIA-1_DSS Installation installs the Land Cover Summarizer Tool in the MapWindow Plugins folder.
(This is usually located at C:\Program Files\MapWindow\Plugins\ModelManager\Elements").

5. Building

To compile the Land Cover Summarizer, add the files below to a Microsoft Visual Studio .NET 2003 Visual Basic .NET Library Project.
Add the following references to the project:

- DssIntfcLib.dll
- MapWinGIS.ocx
- MapWinInterfaces.dll
- stdole.dll

- System.dll
- System.Data.dll
- System.Drawing.dll
- System.Windows.Forms.dll
- System.Xml.dll

You are now ready to compile the project by clicking the Build Solution menu option under the menu Build.

LandCoverSummarizer project files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
Common.vb	Common functions used by the land cover changer.
DBClient.vb	Contains the routines used to connect to a database and store the computed land cover summary.
frmConfiguration.vb	Contains the routines used to select a land cover summary database.
frmModelProps.vb	Contains the routines used to select the settings for the Land Cover Summarizer.
frmSelectLayer.vb	Contains the routines used to select a layer from MapWindow's view.
LandCoverFilter.vb	Implements MapWindow Plugin and DSS Interface routines, allowing Land Cover Summarizer to act as a MapWindow Plugin and a DSS element.

The Land Cover Summarizer is installed in all versions of the WRIA-1 DSS installation.

Technical Documentation: Macroinvertebrate Data Viewer

Last Revision: 06/15/06

Table of Contents

1. *Table Design*
   - *Schema, Keys, Indexes and Structure*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*

3. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*

1. Table Design

Following is a list of tables that must be included with the Macroinvertebrate Data Viewer Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters. This database must be an Access or *.mdb database.

Table: Group_Identification
Description: Contains the Group_ID, Taxa, and Sorting Crew for each of the macroinvertebrate samples.

Field	Type	Size	Key Field	Primary Key	Comments
Group_ID	Text	50	Yes (Duplicates OK)	No	This is an ID for each Group that processed a sample. It corresponds with the Group_ID Field in the Macroinvertebrate_Data table. NOTE: *INSE used the list with numbers, while Bellingham used the list with letters.*
Taxa	Text	50	No	No	Name of the bugs that define this group
Sorted By	Text	50	No	No	Indicates which crew sorted the original sample

Table: Macroinvertebrate_Data
Description: Contains the MacroInvertebrate Sample data for each Site, Group pair for the sizing groups = 1-2mm, 2-4mm, 4-6mm, 6-8mm, 8-10mm, >10mm.

Field	Type	Size	Key Field	Primary Key	Comments
Site_ID	Number	Double	Yes (Duplicates OK)	No	Site Identifier for this sample. It corresponds with the Site_ID Field in the Site_Information table and the Unique_SiteID_List table.
Group_ID	Text	50	No	No	Group Identifier for this sample. It corresponds with the Group_ID Field in the Group_Identification table.
Size1_2_mm	Number	Double	No	No	Sample count for the Group Size = 1 – 2mm. The Default Value = 0 for this field.
Size2_4_mm	Number	Double	No	No	Sample count for the Group Size = 2 – 4mm. The Default Value = 0 for this field.
Size4_6_mm	Number	Double	No	No	Sample count for the Group Size = 4 – 6mm. The Default Value = 0 for this

					field.
Size6_8_mm	Number	Double	No	No	Sample count for the Group Size = 6 – 8mm. The Default Value = 0 for this field.
Size8_10_mm	Number	Double	No	No	Sample count for the Group Size = 8 – 10mm. The Default Value = 0 for this field.
Size10_+_mm	Number	Double	No	No	Sample count for the Group Size = >10mm. The Default Value = 0 for this field.

Table: Site_Information

Description: Contains all the information (Site ID, Site Name, Sample Name, Sample Type, Time In and Out, Velocity In and Out, Site Location, Net size, etc.) about a Site for each sample that was taken and processed.

Field	Type	Size	Key Field	Primary Key	Comments
Site_ID	Number	Double	Yes (Duplicates OK)	No	This is a unique ID for each Site where a sample was taken. It corresponds with the Site_ID Field in the Macroinvertebrate_Data table and the Unique_SiteID_List table.
Site_Name	Text	255	No	No	This is the Name of the Site where the sample was taken. It corresponds with the Site_Name Field in the Unique_SiteID_List table.
Sample_Name	Text	255	No	No	This is the Name for the sample that was taken. It corresponds with the Sample_Name Field in the Unique_SiteID_List table.
Fraction_Subsampled	Number	Double	No	No	This is the fraction of the sample that was actually counted. *For Example: A value = 0.5 would mean that 50% or 1/2 of the sample was counted.*
Sample_Type	Text	255	No	No	This is the Type of sample that was taken. There are two types available : *drift* and *benthic*
Initials	Text	255	No	No	These are the initials of the technician who dealt with the sample in house. *NOTE: The technician did not necessarily collect the sample.*

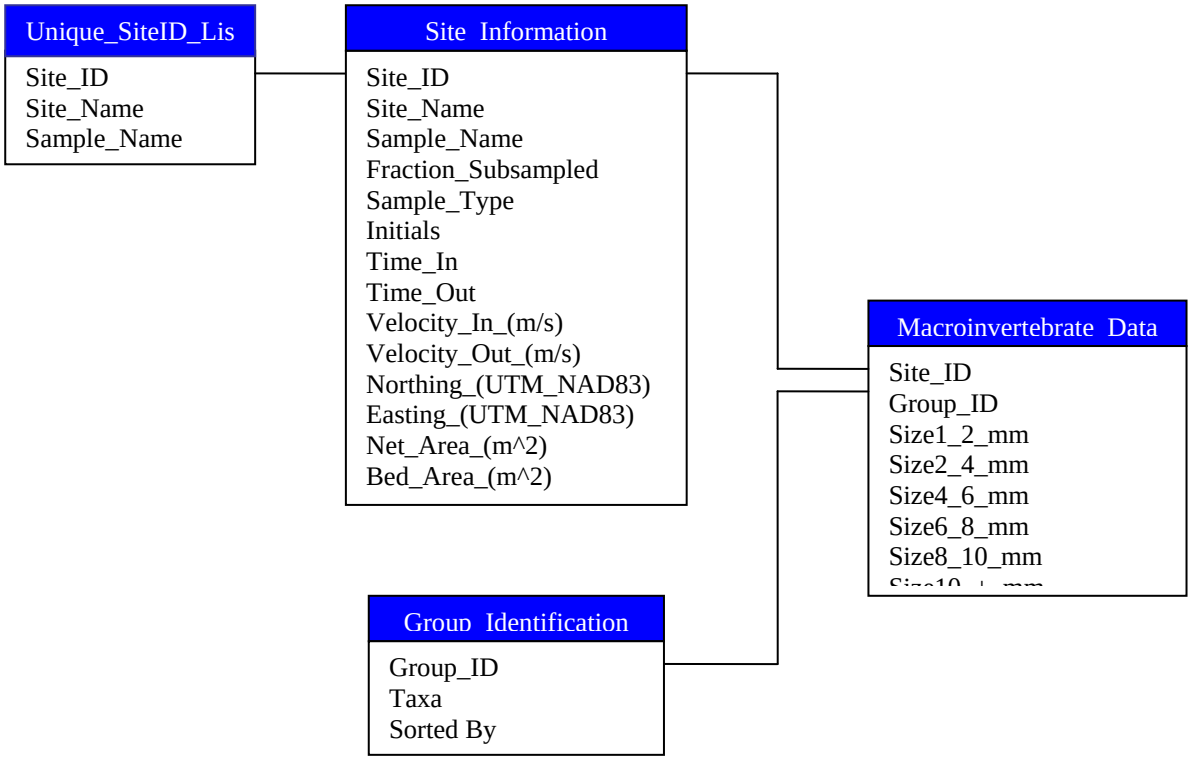
Time_In	Text	255	No	No	This is the Time that the net was put into the water to collect the sample. Value is in Military Time (0:00 – 24:00). *NOTE: this value is only valid if Sample_Type = drift.*
Time_Out	Text	255	No	No	This is the Time that the net was taken out of the water when collecting the sample. Value is in Military Time (0:00 – 24:00). *NOTE: this value is only valid if Sample_Type = drift.*
Velocity_In_(m/s)	Number	Double	No	No	This is the Velocity of the water when the net was put into the water to collect the sample. Value is in meters per second (m/s). *NOTE: this value is only valid if Sample_Type = drift.*
Velocity_Out_(m/s)	Number	Double	No	No	This is the Velocity of the water when the net was taken out of the water when collecting the sample. Value is in meters per second (m/s). *NOTE: this value is only valid if Sample_Type = drift.*
Northing_(UTM_NAD83)	Number	Double	No	No	This is the Northing value (Y-value) of the location of the Site where the sample was taken. Value is in UTM_NAD83 coordinate projection. *NOTE: this is NOT the location of the net or benthic device used to collect the sample.*
Easting_(UTM_NAD83)	Number	Double	No	No	This is the Easting value (X-value) of the location of the Site where the sample was taken. Value is in UTM_NAD83 coordinate projection. *NOTE: this is NOT the location of the net or benthic device used to collect the sample.*
Net_Area_(m^2)	Number	Double	No	No	This is the Area of the Net that was used to collect the sample. Value is in square meters (m^2). *NOTE: this value is only valid if Sample_Type = drift*
Bed_Area_(	Text	255	No	No	This is the Area enclosed by the

m^2)					benthic device that was used to collect the sample. Value is in square meters (m^2). *NOTE: this value is only valid if Sample_Type = benthic.*
Net_Meshsize_(um)	Number	Double	No	No	This is the Size of the mesh that is in the net or benthic device used to collect the sample. Value is in micrometers (um).

Table: Unique_SiteID_List

Description: Contains the unique list of Site, Sample pairs available.

Field	Type	Size	Key Field	Primary Key	Comments
Site_ID	Number	Integer	Yes (Duplicates OK)	No	This is a unique ID for each Site, Sample pair. It corresponds with the Site_ID Field in the Macroinvertebrate_Data table and the Site_Information table.
Site_Name	Text	255	No	No	This is the Name of the Site where the sample was taken. It corresponds with the Site_Name Field in Site_Information table.
Sample_Name	Text	255	No	No	This is the Name for the sample that was taken. It corresponds with the Sample_Name Field in the Site_Information table.



2. Data Needs

The only data that is required for the MacroInvertebrate Viewer is the Access database with the above defined table design. The database connection is only set up for an Access database. The associated database should be located in the

C:\Program Files\WRIA-1_DSS\DSS_Data\Macroinvert Data

file folder.

3. Dependencies

The Macroinvertebrate Data Viewer plug-in requires the following software components and modules to be installed:

Software:

MapWindow 3.1

Visual Studio .NET 2003

Modules and Components:
MapWindow Interfaces
Microsoft Chart Control 6.0 (SP4) (OLEDB)
MSDATARC

Integrated Development Environment (IDE)
Visual Studio .NET 2003

4. Setup

To begin using the Macroinvertebrate Data Viewer, it must first be loaded into MapWindow. Once it is loaded, click on the bug icon, , located on the Map Window toolbar If the database associations (as described in *Section 2: Data Needs*) have been configured properly and the database is accessible, then the Macroinvertebrate Data Viewer will appear allowing the user to select and view macroinvertebrate sample data.

If the data associations have not been properly set, then a Connection Form will appear allowing you to type a path to or select the database. Then, after the database has been properly associated, select the bug icon, , from the Map Window toolbar and the viewer will appear allowing the user to view the collected data.

See the User's Manual for more information and details on associating the data with the Macroinvertebrate Data Viewer.

5. Code Compiling

Compiling the Macroinvertebrate Data Viewer is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 3: Dependencies* are present, load the project into Visual Studio .Net 2003. This Plugin was created using C# (C Sharp).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.cs	Contains information relating to the DLL assembly. Generated by C#.NET.
bug.ico	Bug icon that is used in the MapWindow Toolbar and on the Macroinvertebrate Data Viewer. *NOTE: this file needs to be an embedded resource*
dbManager.cs	Class that contains variables and functions to access and read the sample data stored in the associated database.
frmDBConnection.cs, frmDBConnection.resx	Displays the Connection form that allows the user to associate the access database with the Macroinvertebrate Data Viewer plug-in.

frmErrorDialog.cs, frmErrorDialog.resx	Displays the form to display an Error Message when an error occurs in the Plugin. It allows the user to view the specific details of an error if they desire.
frmMacroDataViewer.cs, frmMacroDataViewer.resx	Displays the Macroinvertebrate Data Viewer form. You show the viewer by selecting the bug icon, , from the Map Window toolbar. This form contains plots, tables to display the data contained in the associated database.
Globals.cs	Contains a class that contains variables (for Map components), properties, and functions that can be used throughout the project.
main.cs	Contains a class that implements the MapWindow plugin interface.
PluginInfo.cs	Contains a class that implements an interface to access (read from and write to) the Project File.

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

AxMSChart20Lib	CopyLocal = True
MapWinInterfaces	CopyLocal = False
MSChart20Lib	CopyLocal = True
MSDATASRC	CopyLocal = False
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
system.web.services	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwMacroInvertDataViewer.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwMIVViewer* subdirectory in the *\Plugins* folder. Next time that MapWindow is run, if the mwMIVViewer.dll was built to the correct folder, the updated changes to the Macroinvertebrate Data Viewer will be available.

Technical Documentation:
Model Manager / Scenario Builder / DSS

Last Revision: 12/30/04

Table of Contents

1. *Table Designs*
   - *Schema, Keys, Indexes and Structure*
   - *Table Descriptions and Notes*

2. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

3. *Setup*
   - *Setup and Configuration*
   - *Interactions between Client and Server*

4. *Building*
   - *Code Compiling*

5. *Model Manager Elements*
   - *Model Manager Interface*

1. Table Designs

Database: DSSData.mdb

Table: FilePaths
Description: This table stores the locations of the files used by models running inside the model manager. A model can request a file of a specific DataType, which will present the user with a dialog asking which of the matching datatypes should be used. They're differentiated by FileID, which can be any identifier – string, number, etc. ScenarioID refers to the scenario which added the file path to the database – scenario 0 is reserved for unedited, base data.

Field	Type	Size	Indexed	Primary Key	Comments
RecordID	Numeric		Yes	Yes	Internal record identifier.
FileID	Text	50	Yes	No	Identifier for this file. For example, "Commercial Zoning", "Comprehensive Zoning"
FilePath	Text	255	No	No	This is the full path to the file on disk.
ScenarioID	Numeric		Yes	No	The ID of the scenario which added this file path. 0 is reserved for unedited base data.
DataType	Text	25	No	No	Type of data. Can be any string, for example "zoning", "land use"

Table: scenario_logs
Description: This table stores the run log generated by a scenario when it is executed. The log contains information about what elements were run and with what parameters.

Field	Type	Size	Indexed	Primary Key	Comments
ScenarioLogID	Numeric		Yes	Yes	Unique identifier for this scenario log.
RunDate	Date/Time		No	No	The date that the run was executed
UserName	Text	50	No	No	The name of the user logged in while running this scenario.
StartDate	Date/Time		No	No	The start date for the date range of the run. Not all elements use this.
EndDate	Date/Time		No	No	The end date for the date range for this scenario run. Not all elements use this.
MapWinVersion	Text	50	No	No	The version of MapWindow in use for this run.
OSName	Text	50	No	No	The operating system this run was executed on.

OSVersion	Text	50	No	No	The version of the operating system this run was executed on.
DSSVersion	Text	50	No	No	The version of the Model Manager in use for the run.
SysRAM	Text	50	No	No	The amount of RAM in the system running this scenario.
SysSpeed	Text	50	No	No	The speed of the system running this scenario.
SysOther	Text	50	No	No	Used for other information about the computer which ran this scenario.
DataVersion	Text	50	No	No	The version of the GIS data in use for the run.
ScenarioLog	Text	0	No	No	The textual log for the scenario run, including the results of each individual element that was run.
ControlFileID	Numeric		Yes	No	This is the scenario

Table: ScenarioControlFiles
Description: This table holds the actual scenarios themselves; when the user saves the scenario, it goes into this table – opening a scenario similarly lists the scenarios in this table.

Field	Type	Size	Indexed	Primary Key	Comments
ScenarioID	Numeric		Yes	Yes	The ID number for this scenario, referenced by all other tables in this database.
ScenarioName	Text	50	Yes	No	The textual name of the scenario.
FileData	Binary	0	No	No	This is the actual scenario data, in a proprietary binary format.
ScenarioGUID	Text	50	Yes	No	The GUID of the scenario – a type of unique identifier.

2. Dependencies

The Model Manager, aka Scenario Builder, aka “DSS”, requires the following software components and modules to be installed:

Software:
Visual Studio .NET 2002

Components:

Microsoft Common Controls 6.0
Microsoft Common Controls-2 6.0
Microsoft Common Controls-3 6.0
Microsoft FlexGrid Control 6.0
MDAC (Microsoft Data Access Components) 2.7
AddFlow ActiveX Component Version 3.0
DotNetBar Component for Menus and Dockable Toolbars Version 2.6.0.1

3. Setup

After building the plugin DLL, create a folder called ModelManager inside the Plugins folder, which may be found wherever MapWindow is installed (typically c:\Program Files\MapWindow). Copy the compiled DLL and any other files produced by the compilation in this directory.

In this folder (ModelManager), create a folder called Elements. Inside this folder, place any compiled model element DLLs you wish to use with the Model Manager.

Start MapWindow, and start the DSS / Scenario Builder as indicated by the user's manual. You will likely be presented with the database configuration screen, which you will need to use to locate the DSSData.mdb file on your hard disk (or locate a DSSData SQL Server database). If you have installed a complete pre-assembled DSS, this will probably be set already. The model manager will save its configuration when the proper database has been located. The Model Manager will then be ready for use.

4. Building

Compiling the Model Manager is a simple process, after ensuring that all of the components listed in section 2 are present. Open the mwDSS.sln project with Visual Studio, and select the Build menu; then select Build All.

The Model Manager Visual Studio project is described below:

mwDSS Model Manager / Scenario Builder Project

File Name	Purpose
DataManager\DataManager.vb	Automates the connection to the underlying DSS database.
Images\	Contains various images used within forms, such as the start and stop images for scenarios.
ScenarioBuilder\AddFlowWrapper.vb	This is the container for the ActiveX addflow component, instantiated once for each open

	scenario.
ScenarioBuilder\frmAuthenticate.vb	Used to set the scenario password.
ScenarioBuilder\frmConfiguration.vb	This form is used to configure the database connection for the scenario builder.
ScenarioBuilder\frmEditBaseData.vb	This allows editing of the file paths for the base data stored in the underlying DSS database.
ScenarioBuilder\frmLogViewer.vb	This provides a user interface to view the scenario run log generated by a scenario run.
ScenarioBuilder\frmNodeProps.vb	Displays the properties of the model element or node currently selected in the scenario.
ScenarioBuilder\ frmScenarioBuilder.DockManager.xml	Settings file controlling the menu items and the attributes of those items within the scenario builder. Read by the DotNetBar ActiveX control.
ScenarioBuilder\frmScenarioBuilder.vb	Main scenario builder form. Utilizes DotNetBar and AddFlow (through the wrapper) to put together the form on initialization.
ScenarioBuilder\frmScenarioProps.vb	Shows and allows editing of the scenario properties, such as name and comments.
ScenarioBuilder\frmSelectControlFile.vb	This is essentially the “open scenario” dialog box.
ScenarioBuilder\frmSelectFileID.vb	This is essentially a “select layer” dialog box, retrieving the options from the file paths stored in the underlying database.
ScenarioBuilder\ModelInfo.vb	A class used to hold the attributes for each model in the scenario builder.
ScenarioBuilder\ModelLoader.vb	This is the class used to load the models from their DLL files, creating a ModelInfo class for each.
Utils\Common.vb	Contains miscellaneous common functions such as swap, append/delete attribute from XML file, etc.
Utils\DBClient.vb	Provides a layer of abstraction to access the database interface.
Utils\FolderUtils.vb	Provides the “browse for folder” dialog box as well as the CopyFolder function.
Utils\frmSelectLayer.vb	This dialog allows the user to select a layer from a file or from layers currently loaded in MapWindow.
Utils\SystemInfo.vb	This class retrieves information about the system, for storage into the scenario run log.
AssemblyInfo.vb	Contains versioning information for this product.
DSSPlugin.vb	This is where the MapWindow plug-in interface is implemented, typing the scenario builder into MapWindow.

5. Model Manger Elements

Model Manager elements will need to implement the model interface. This is a .NET DLL interface very similar to the MapWindow interface. A model manager element will compile to a DLL, and, assuming it properly implements the model interface and is placed inside the Elements folder (located where the Model Manager DLL is located), it will appear as an element inside the Model Manager.

It's useful to note that a model element can implement both the MapWindow plug-in interface and the Model Manager interface at the same time. Any other interface which is useful to implement may also be implemented. If both MapWindow and Model Manager interfaces are implemented, some functions will overlap, such as "Author", "BuildDate", and "Version". If your programming language supports it, you may have one function implement the function from both interfaces at the same time. In VB.NET, this is accomplished by listing all desired functions to be implemented, for example: "string Author implements DssIntfc.IDssModel.Author, MapWinInterfaces.IPlugin.Author" (Exact names of the interfaces may change slightly.)

The Model Interface is described below:

Model Manager Interface

The following functions need to be implemented in any scenario element intended to be used within the model manager. More detail and examples are given in a separate document entitled "Programmer's Tutorial: How to Create a DSS Model". The functions are shown in C syntax:

return_type function_name (parameter1_type parameter1_name, parameter2_type parameter2_name, …)

string RevisionNotes

Provides notes about the scenario element. This information will appear on the "scenario properties" dialog. This is a "get" property (cannot be set). So, the Microsoft C# .NET code would look something like:

```
public string RevisionNotes
{
    get
    {
        return "This is a simple example of a get property.";
    }
}
```

This is the same syntax that should be followed with any other "get" property. See the reference for your particular programming language to see how to implement get properties. For most languages, it will suffice to return the appropriate value in the function body.

bool Execute(DssIntfcLib.IDssManager DssManager, int TimeStep, string DataPath, DateTime StartDate, DateTime EndDate, string ScenarioID)

The Model Manager calls this function to initiate the execution of the model. All parameters and data that have been configured as a consequence of the ShowDialog function (below) will have been loaded and made ready before this function is called.

Parameters:

DSSManager: This is a reference to the Model Manager that called the execute function.

TimeStep: This is the time step for execution in seconds. For example, 3600 would mean a timestep occurs every 5 minutes.

DataPath: This is the path where data should be written if the element needs to write directly to disk.

StartDate: This is the start date chosen with the date range selectors in the model manager interface. This may be ignored if your scenario element does not use a date range.

End Date: This is the end date chosen as above. This also may be ignored if you don't need to restrict operations to a particular date range.

ScenarioID: This is the scenario ID of the scenario which has called the execute function. This is provided as a way for the element to mark any output, for example naming a file with the scenario ID, so that the end user can tell what scenario produced the file.

int ParamCount

This is a "get" property which should return the number of data items that your scenario element needs to save and retrieve as a part of the scenario into which the element is embedded. See also get_ParamValue, set_ParamValue, and get_ParamName.

DssIntfcLib.DssModelType ModelType

This is a "get" property which will return what type of model this is. The options are "filter", "model", "data editor", or "other". There are no rules specifying what makes a scenario element any one of these; it's only intended for organizational purposes. The only thing this will affect is which tab the element will appear on inside the model manager.

string BuildDate

This is a "get" property which will return the date that the element was last compiled.

string Description

This is a “get” property which will return a textual description of what the scenario element does. This will appear on the “element properties” dialog.

string Author

This is a “get” property which will return a string identifying the author of the scenario element. This may be any string – a name, an organization, or any text.

bool ShowDialog(bool Locked, DssIntfcLib.IDssManager DssManager, string DataPath)

This function is called when the scenario element is double-clicked within the model manager. This function should show a form with any configuration options or settings which need to be configured for the model to execute. The function should save the settings retrieved from the user into the properties which will be set and retrieved using set_ParamValue, get_ParamValue, get_ParamName and ParamCount. The function does not need to worry about saving the settings anywhere – so long as the settings are accessible using the functions just named, the model manager will deal with saving and restoring these values automatically.

Parameters:

Locked: This indicates that the scenario is locked, and changes should not be allowed. Settings may still be viewed, however.
DssManager: This is a reference to the instance of the model manager which has called this function.
DataPath: This is the path that the scenario element should use if the element needs to read or write directly to data on disk. This should be avoided in favor of the automatic parameter saving system.

string Name

This is a “get” parameter which should return the name of the scenario element as it will be displayed in the scenario builder. This should be fairly short.

string Version

This is a “get” parameter which should return the version of the scenario element.

void set_ParamValue(int Index, string pVal)

This function is called by the model manager when previously saved values are being restored to the element. The model manager will call this function as many times as the function ParamCount returns, also providing the appropriate value for each parameter. These values should be saved into appropriate variables in the

scenario. The index provided to this function will match the index used by get_ParamValue and get_ParamName.

Parameters:

Index: This is the identifier for the value being restored into the element.
pVal: This is the value to be restored to the element. Note that this is a string; this doesn't mean that only string data can be saved and restored: non-string data can be marshaled into a string format and saved.

string get_ParamValue(int Index)

This function is called by the model manager when the scenario is saved. The function should return the value for the parameter indicated by Index, which will then be saved by the model manager. Like set_ParamValue, the function will be called as many times as the function ParamCount returns. If ParamCount returns 2, the function will be called twice to get both values.

Parameters:

Index: This is the identifier for the value being saved from within the element.

object ToolBoxIcon

This is a "get" parameter which should return the image which will be displayed as the icon for this element inside the model manager toolbox. The image can be a bitmap or icon, and should be 32 pixels by 32 pixels. A default image will be used if the image provided is invalid.

string get_ParamName(int Index)

This function should return the name of the parameter referenced by Index. This function yields a name for the Index values used by get_ParamValue and set_ParamValue.

string ResultsSummary

This function is a "get" parameter which should return a summary of the execution of the model. The model manager will call this function after calling Execute, to get the results of the execution. The function should return a string specifying any input files that were used and any output files that were created, and the string should also summarize any actions taken during execution. The summary should also contain any error string which may have been generated. The text returned by this function will appear in the scenario log.

Some of the functions in the Model Manager return an object called IDssManager. This is a reference to the Model Manager that's currently running your scenario element. The methods available on this object are as follows: (again, shown in C syntax.)

HRESULT ReportProgress(IDssModel * sender, BSTR Message, long PercentDone)

This function is used to report the progress of your scenario element's execution to the Scenario Builder; the percentage done will be reflected in the progress bar along with the message you give.

The HRESULT return may be ignored in most languages.

Parameters:

Sender: This is a reference or a pointer to your scenario element itself.

Message: This is the progress message to be displayed on your progress bar.
High-level languages like most Microsoft .NET languages will interpret the BSTR type as a simple "String" type. Low-level languages like C will need to pass a B-string.

PercentDone: This is the percent done expressed numerically. This will be displayed on the progress bar.

HRESULT ReportError(IDssModel * sender, BSTR Message)

This function is used to report an error during execution to the Scenario Builder. This will appear in the scenario run log.

Parameters:

Sender: This is a reference or a pointer to your scenario element itself.

Message: This is the error message to be placed in the run log.
High-level languages like most Microsoft .NET languages will interpret the BSTR type as a simple "String" type. Low-level languages like C will need to pass a B-string.

HRESULT CancelWasRequested(VARIANT_BOOL *result)

This function is used to ask the scenario builder if a cancel has been requested (by the user clicking the stop button in the scenario builder). This should be checked periodically during your Execute function if you're doing complex calculations; the model manager will never halt your scenario element, so it's your responsibility to check to see if the run has been cancelled and stop processing nicely.

Parameters:

Result: This should be a pointer or a reference to a Boolean data type. This will be set by the function to True or False to indicate whether a cancel has been requested.

HRESULT ReportDialogClosing(IDssModel * sender, VARIANT_BOOL SaveParameters)

This function is used to report to the scenario builder that a dialog box is closing. Typically, dialog boxes in scenario elements are used to gather data – so, this is important to call at dialog closing because signaling that a dialog is closing tells the model manager that it's time to save the parameters in your scenario element (using the interface functions get_ParamValue, set_ParamValue, get_ParamName, and ParamCount). Therefore, make sure that you've set the variables which get_ParamValue is returning before calling this function.

Parameters:

Sender: This is a reference or a pointer to your scenario element itself.

SaveParameters: This tells the Scenario Builder whether you'd like your scenario element's parameters saved (using the interface functions mentioned above). Most high-level languages can pass any Boolean data type into this function.

HRESULT GetFilePath(BSTR FileID, BSTR * result)

This function is used to get the path on disk to the file identified by the string FileID. This is used in tandem with the function ShowSelectFileDialog.

Parameters:

FileID: This is the string uniquely identifying the file whose path you need.

Result: This should be given a pointer or a reference to a string where the path to the requested file can be written. In other words, the string you provide to Result will contain the path to the file you've requested after the function call.

HRESULT AddFilePath(BSTR FileID, BSTR FilePath, BSTR DataType, VARIANT_BOOL * result)

This function is used to add information about a file to the Scenario Builder's database for later retrieval using GetFilePath or ShowSelectFileDialog.

Parameters:

FileID: This is the unique string identifier for this file. (Most high-level languages, like VB.NET, can pass a "String" data type to BSTR parameters).

FilePath: This is the full path to the file on disk.

DataType: This is a string representing the type of data that the file is. The string may be anything you desire – e.g., "Land Cover Data", "Access Database", etc.

Result: A reference or pointer to a Boolean data type should be provided for this parameter. After the function call, the variable passed as this parameter will contain True or False depending on whether the call was successful.

HRESULT ShowSelectFileDialog(BSTR AllowedDataType, BSTR * result)

This function will display a dialog to the user with a list of files which match the parameter AllowedDataType. The user will see a list of the FileIDs for the files matching the data type.

Parameters:

AllowedDataType: This is the string representing the type of data that you wish the user to select. See *DataType* under the function AddFilePath; these two parameters correspond.

Result: A reference or pointer to a string data type should be passed as this argument. After the function call completes, the variable passed as this parameter will contain the FileID of the file that the user selected in the dialog.

HRESULT GetScenarioFilePath(BSTR ScenarioID, BSTR FileID, BSTR * result)

This function returns the path to the directory where scenario data is being stored.

Parameters:

ScenarioID: The scenario ID for which you'd like the path to data.

FileID: The FileID of the file which you'd like the data path for.

Result: A reference or pointer to a string data type should be passed as

this argument. After the function call completes, the variable passed as this parameter will contain the path to the data.

HRESULT MinimizeModelManager()

This function will cause the scenario builder screen to minimize, allowing any windows below (such as MapWindow) to become visible. This is useful when a scenario element is implementing both the MapWindow interface and the Model interface, and the scenario element wishes to use the MapWindow map to collect information or interact with the user.

HRESULT MaximizeModelManager()

This function will cause the scenario builder screen to restore itself if it has been minimized. If MinimizeModelManager has been called, then this should eventually be called as well.

Appendix

Programmer Tutorial:

How to create a DSS Model for the MapWindow 3.0 DSS Model Manager

Darrel Brown
9/18/2003

Table of Contents

Index:

Introduction 119
Project creation 119
Implement Interface(s) 122
Basic model properties 122
Inputs, Outputs and Parameters 123
Filling in the details 124

Table of Figures:

Figure 1- Creating a new Class Library project. 120
Figure 2 - Add references 120
Figure 3 - Add Reference dialog 121
Figure 4 - Dual implementation 122
Figure 5 - Adding parameters dialog 125
Figure 6 - Completed parameters page layout 125
Figure 7 - Validation Function 126
Figure 8 – Parameter access properties 127
Figure 8 – ShowDialog function 128
Figure 8 – Declaration of model variables 129
Figure 9 – Execute function 130
Figure 10 – Results summary 130

1. Introduction

In this tutorial paper, I will discuss how to create a DSS model plug-in. The model plug-in can be written in any language that supports COM. For this tutorial, I will give examples in Visual Basic.net (VB.net) because I think that VB.net illustrates the concepts required to create the plug-in more clearly than the other popular alternatives. Some alternatives are: Microsoft Visual C++, Microsoft C#, Microsoft Visual Basic 6, Borland C++ Builder and Borland Delphi. I have found that implementing COM interfaces is easiest to do in Visual Basic (6 or .net) and C#.

Before you can create a DSS Model Manager plug-in, you must have the MapWindow and DSS Model Manager installed. You can get the MapWindow from http://www.mapwindow.com if you do not already have it.

Before I begin the details of this tutorial, I must emphasize again that I am using VB.net for reasons of clarity and personal preference. You may use any language that supports COM.

2. Project creation

In Visual Studio.net, create a new project. Create the project as a Class Library. Give the project a name that represents the function of the model. In my case I chose to call this model "SampleModel." After you press OK, change the name of your class from Class1 to something more descriptive. For this tutorial I called my class "MyModel." Ideally this class name describes your model in some way.

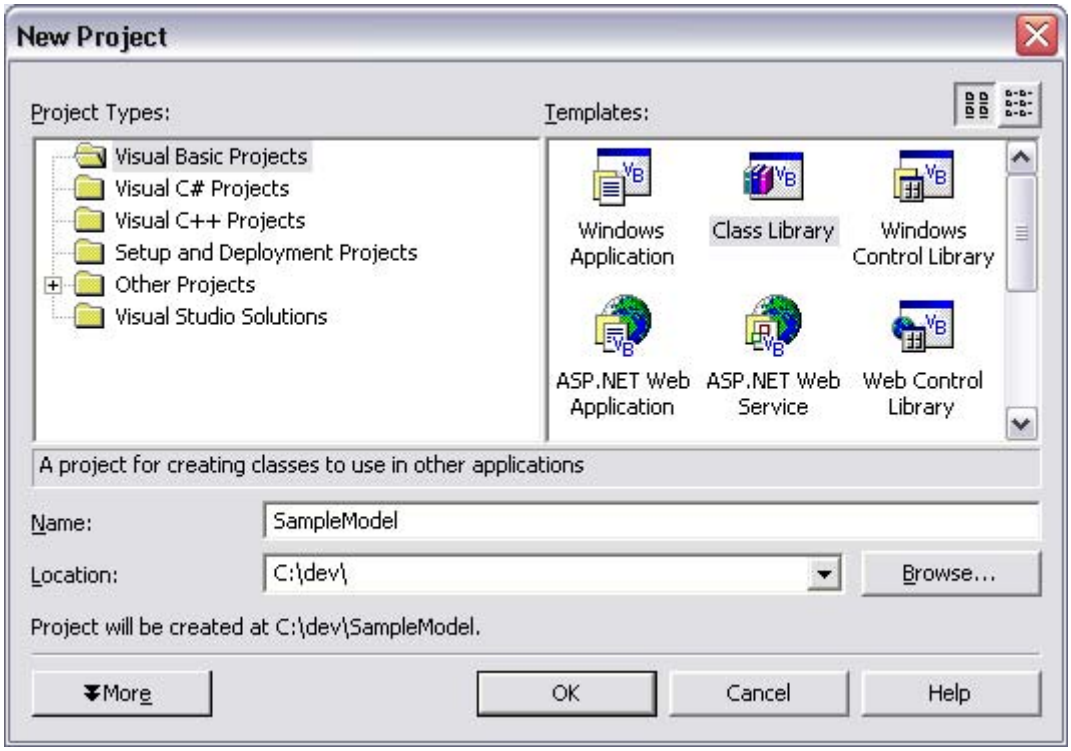


Figure 1- Creating a new Class Library project.

The first step is to add references to the MapWindow Interfaces and DSS Model Manager Interfaces. The easiest way to do this is to right-click on the references item in the Solution Explorer, then select Add Reference.

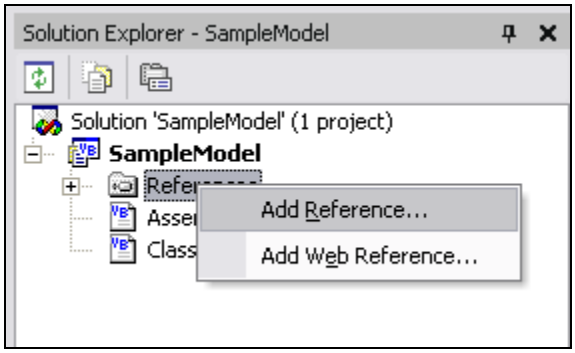


Figure 2 - Add references

It doesn't matter which reference you add first, but since the Add Reference dialog defaults to .NET references first, I'll add the MapWinInterfaces reference first. This reference will not show up automatically in the list. You must browse to the file in order to add it. In most cases, the needed interface is installed at "C:\Program Files\MapWindow3.0\MapWinInterfaces.dll" unless the file was installed in another location.

The next reference is listed on the COM tab. The reference needed is called “DSS Interface Definitions 1.0.” Select this reference then press OK. These references contain the interfaces used to access the Model Manager and the MapWindow. (See Figure 3)

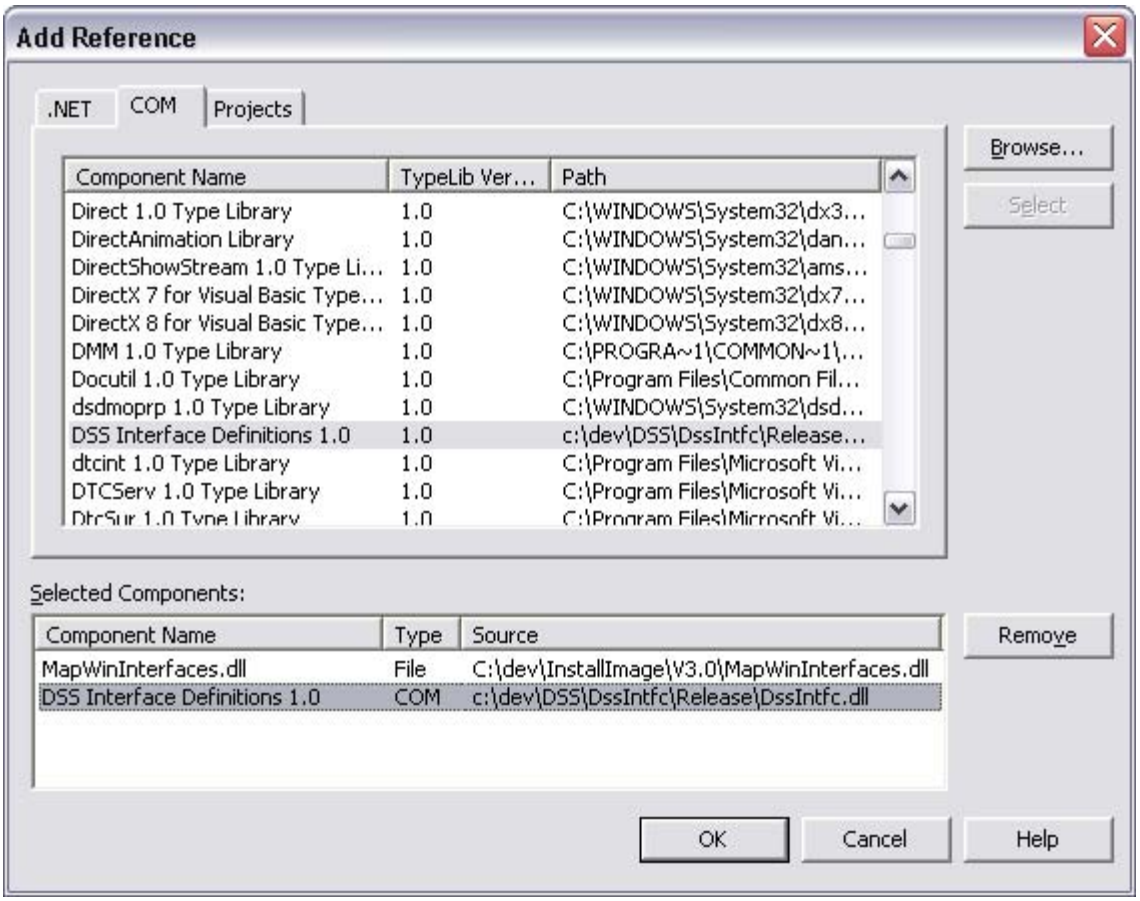


Figure 3 - Add Reference dialog

3. Implement Interface(s)

In order for the Model Manager to recognize your program as a model, you must implement the IDssModel interface located in the DssIntfcLib namespace. If you are not sure what this means then perhaps it would be a good time to look up the Implements statement in the Visual Studio documentation. Implement all of the methods and properties from the IDssModel interface. You can leave all of the details blank for now. We will get to them shortly.

If your model must interact with the MapWindow, it must also implement the IPlugin interface in the MapWindow.Interfaces namespace. All of the methods must be implemented from this interface also. Note, there are some properties that are shared between the IDssModel and IPlugin interfaces. In these cases it is easiest to have one property declaration that handles both interfaces, illustrated in Figure 4.

```
Public ReadOnly Property Author() As String _
    Implements DssIntfcLib.IDssModel.Author, _
    MapWindow.Interfaces.IPlugin.Author
    Get
        Return "Utah State University"
    End Get
End Property
```

Figure 4 - Dual implementation

Only having one property that returns the values shared by both interfaces reduces code duplication and makes it easier to keep your values synchronized. The properties that are shared are:

- Author
- BuildDate
- Description
- Name
- Version

For the purposes of this tutorial, I am going to keep it simple, implementing only the IDssModel interface.

Make sure that all methods and properties for the interfaces are implemented. If you forget one, the Implements statement at the top of the class will be underlined. Another way to make sure that you have implemented the interface completely is to build the project. If there are any errors, the compiler will notify you.

Basic model properties:

Now is a good time to go through the Author, BuildDate, Description, Name, Revision Notes and Version properties, returning the values you wish the user to see on the Model Properties dialog in the Model Manager.

The ModelType property determines which toolbox your model will be dropped in. This property has very little effect on how the model is treated but it does help to organize models by function. If your model simply changes data into a format that can be recognized by another model then it should be called a Filter. If your model modifies data but does not do any real analysis on the data then it should be a Data Editor. These categories are not enforced in any way, so it is really up to you to decide how you wish your model to be classified.

Along with the ModelType, you can also define a custom icon to override the default icon used in the Toolbox and Layout. This property will accept every standard image type used in Visual Studio. The supported image types are Icon, Bitmap and StdPicture. The model manager resizes the icon to make it the same size as all the other icons in the Toolbox and on the Layout.

4. Inputs, Outputs and Parameters

Before we can discuss how to implement inputs, outputs and parameters it is very important to understand the difference between them and how the Model Manager handles each of them.

Inputs and outputs are text strings that are stored in the DSS database using a "key" value. The text string can be the path to a file that used by the model while executing. This way, when the model runs it can request an input from the database using its Key, for example, "LANDCOVER", the DSS return to the model the string value in the database stored with the Key identifier. In the example, it would probably return a file path to the land cover grid. Inputs are also stored with a scenario ID. For the raw data sets, the scenario ID is 0. However when a model runs, it can add a new record to the inputs and outputs table identifying the output data and associating a scenario number with it.

This is best illustrated with an example. Say that your scenario has a land cover data changer followed by a model that uses a land cover grid as an input. When the scenario executes, the land cover data changer is invoked. This data changer requests data from the DSS database stored with the key, "LANDCOVER". The DSS knows that we are in the middle of a scenario run, for example, #36. When the land cover changer requests the LANDCOVER data set, the DSS looks in the table for an entry with the key, "LANDCOVER" and the scenario ID #36. Not finding it, the DSS then looks for an entry with the key, "LANDCOVER" and the scenario ID #0. Finding this entry, it returns the string data to the model. The string data in this case would contain the path and filename to the raw land cover grid file. The land cover data changer then makes specified changes to the land cover grid and saves the changes under a new filename such as "lc36.bgd". It then stores this information as a new entry in the DSS database under the same key name, "LANDCOVER" but with the new file path and the scenario ID that created it (#36).

Next the DSS executes the model. This model requires a land cover grid as an input, so it makes a request to the DSS for this dataset. The DSS repeats the same steps performed previously for the land cover changer, looking in the database for an entry with the Key "LANDCOVER" and the scenario ID #36. Finding this entry, it returns the string value to the model – in this case it returns the string, "lc36.bgd". Now the model has an input data file that has been edited by the

land use changer in the previous step of the same scenario. Doing this allows one to build complex scenarios with many changes to data while passing only pointers to the changed data between scenario elements.

"Parameters" are different than model inputs and outputs and are saved on a per-node basis in the scenario layout meaning that each instance of the model in a Layout can have its own set of parameter values. Parameters are typically modifiers that your model uses when executing. For example, the model mentioned above might compute sediment load from a watershed using the land cover grid as an input. However there are likely parameters that define how the model works and these are editable by the user through a parameter page that is displayed when the user double-clicks on the model element node in the DSS.

Parameters can be used to store data entered by the user for each model instance. When the model node is deleted or the layout is deleted any parameters that were set are discarded.

The sample model used in this tutorial will calculate a simple weighted average. Choosing a simple model makes it easier to see the mechanics of creating a model without cluttering thing up with complicated code. The inputs to this model will be the values to average. The parameters will be the weights for the input values. The output will be the value of the weighted average.

5. Filling in the details

The first step I will take in this tutorial is to create the parameters page. The parameters for this tutorial are two numeric values that are the weights for the weighted average.

To create a parameters page, right click on the title of your Model in the Solution Explorer, then click on *Add*, then *Add Windows Form…* Name your parameter page whatever you like. I named mine frmParameters.

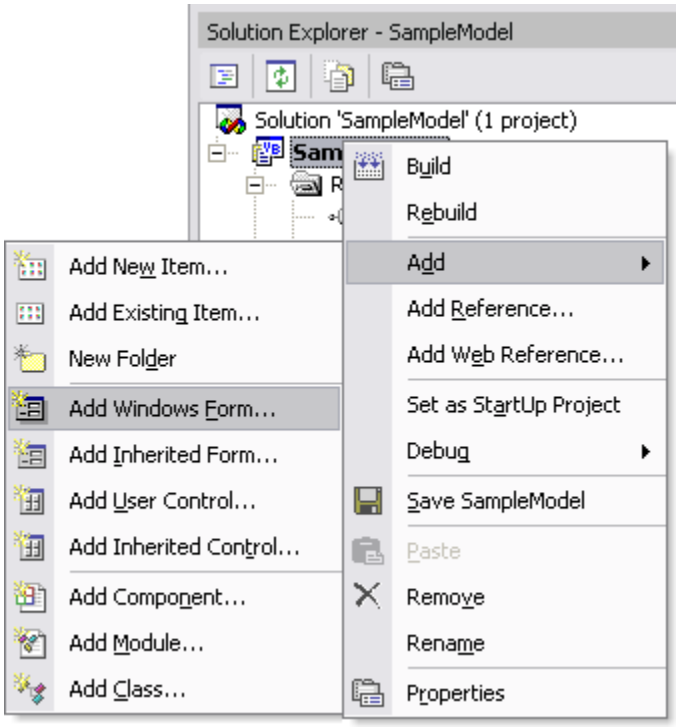


Figure 5 - Adding parameters dialog

Next I added two labels and text boxes for the two weights. I also grouped these in a group box for the sake of appearance. There is also an *OK* and a *Cancel* button. I changed the form's text property to *Sample Model Parameters.* A good suggestion to make your dialogs look and feel more professional is to set the form's `AcceptButton` and `CancelButton` properties. I also set the `DialogResult` properties on the *Accept* and *Cancel* Buttons to be *OK* and *Cancel* respectively. Another nice touch is to change the `FormBorderStyle` better represent the purpose of the form. In this case I decided that a `FixedDialog` style worked best. Since the dialog is not resizable I turned off the `MinimizeBox` and `MaximizeBox`. Finally, I set the `StartPosition` to be `CenterScreen`. Of course, all of these details are up to you, and depend a great deal on your needs.

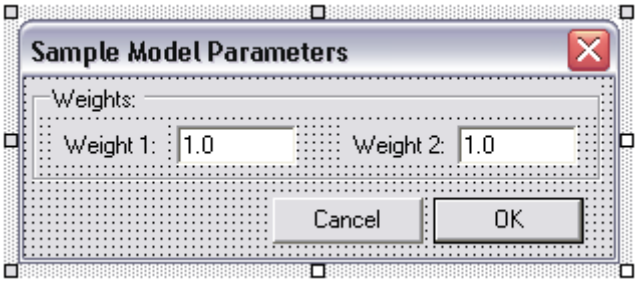


Figure 6 - Completed parameters page layout

The only tasks for the parameters page dialog are to validate the inputs and make them accessible from the model. I will write a validate routine that makes sure the text entered is numeric. One important thing to note is that a layout can be locked and password protected so that the values

on the parameter pages cannot be modified. The model manager passes a Boolean parameter in *ShowDialog* called *Locked*. If this value is true, the model writer should make sure that his/her parameters can be viewed but not changed.

```
    ' The Weight_Validating Method handles the validation
    ' for both of the text boxes
    Private Sub Weight_Validating(ByVal sender As Object, _
                ByVal e As System.ComponentModel.CancelEventArgs) _
                Handles txtWeight1.Validating, txtWeight2.Validating
        ' Make sure to catch all exceptions
        Try
            ' Cast the sender to a text box object.  If this
            ' fails, an exception will be thrown
            Dim txt As System.Windows.Forms.TextBox
            txt = CType(sender, System.Windows.Forms.TextBox)
            If IsNumeric(txt.Text) = False Then
                ' Setting cancel = true makes sure that the user
                ' cannot leave until the problem is fixed
                e.Cancel = True

                ' Just to be nice, we will select all text
                ' to help the user see what went wrong
                txt.SelectAll()

                ' Exit immediately because the validation failed
                Exit Sub
            End If

        Catch ex As System.Exception
            ' There was an error, show a descriptive message
            MsgBox("The following error has occurred:" & vbCrLf & _
                    ex.Message & vbCrLf & ex.StackTrace.ToString(), _
                    MsgBoxStyle.Exclamation Or MsgBoxStyle.OKOnly, _
                    "Error in Weight_Validating")
        End Try
    End Sub
```

Figure 7 - Validation Function

Now that the inputs are properly validated, the only thing left is to make the values accessible to the model.

```
    Public Property Weight1() As Double
        Get
            If IsNumeric(txtWeight1.Text) Then
                Return CDbl(txtWeight1.Text)
            Else
                Return 1.0
            End If
        End Get
        Set(ByVal Value As Double)
            If IsNumeric(Value) Then
                txtWeight1.Text = Value.ToString()
            End If
        End Set
    End Property

    Public Property Weight2() As Double
        Get
            If IsNumeric(txtWeight2.Text) Then
                Return CDbl(txtWeight2.Text)
            Else
                Return 1.0
            End If
        End Get
        Set(ByVal Value As Double)
            If IsNumeric(Value) Then
                txtWeight2.Text = Value.ToString()
            End If
        End Set
    End Property

    Public Property IsLocked() As Boolean
        Get
            Return Not grpWeights.Enabled
        End Get
        Set(ByVal Value As Boolean)
            grpWeights.Enabled = Not Value
        End Set
    End Property
```

Figure 8 – Parameter access properties

The parameter page is now completely done so we need to go back to the model and show the dialog when the user requests it. This is done by the *`ShowDialog`* method that you implemented from the IDssModel interface. There is only one tricky thing about the ShowDialog method: You must tell the Model Manager when you are done showing your dialog. This is necessary because it is possible to have a non-modal dialog that interacts with the MapWindow. A non-modal dialog is one that allows you to interact with other forms while the dialog is open. A modal dialog does not allow the user to interact with anything else until the dialog is closed. To summarize, modal dialogs are synchronous, non-modal dialogs asynchronous. If the dialog is asynchronous you have to have your form notify the model when it closes. This can be done in a number of ways:

- Implement a callback system so that the parameter page itself notifies the model that it is closing

- Create a function in the model class that is a *FormClosing* delegate, and then before you display the dialog, add the model class *FormClosing* delegate function as a handler of the dialog form's *FormClosing* event.

Neither of these two options is really very difficult if you are familiar with delegates or callbacks. If you are not familiar with either of those two concepts it is worthwhile to learn about them because they can in some situations make a task that seems complicated simple. In the tutorial I will use a modal (synchronous) dialog to make things simpler. Note that I it is important to return true. If you do not return true, the Model Manager will not save any parameters because it assumes something went wrong.

```
Public Function ShowDialog(ByVal Locked As Boolean, _
    ByVal DssManager As DssIntfcLib.IDssManager, _
    ByVal DataPath As String) As Boolean _
    Implements DssIntfcLib.IDssModel.ShowDialog

    Try
        ' Create a new instance of the parameters form
        Dim myDialog As New frmParameters()

        ' Load the values from the model
        myDialog.Weight1 = m_Weight1
        myDialog.Weight2 = m_Weight2
        myDialog.IsLocked = Locked

        ' If the user clicks ok then save the values.
        If myDialog.ShowDialog = Windows.Forms.DialogResult.OK Then
            ' Save the values entered on the parameters page
            m_Weight1 = myDialog.Weight1
            m_Weight2 = myDialog.Weight2
            ' Report that my dialog has closed, and request
            ' that the parameters should be saved
            DssManager.ReportDialogClosing(Me, True)
        Else
            ' Don't attempt to save any parameters
            DssManager.ReportDialogClosing(Me, False)
        End If

        ' Clean up the dialog form
        myDialog.Dispose()
        myDialog = Nothing
        Return True
    Catch ex As System.Exception
        MsgBox("An error occurred in ShowDialog:" & vbCrLf & _
               ex.StackTrace.ToString(), , "Error in ShowDialog")
    End Try
End Function
```

Figure 8 – ShowDialog function

You might have noticed that I made references to m_Weight1 and m_Weight2 in the ShowDialog method. These are declared at the top of the model class along with the values from

the inputs that will be used during the model execution. Notice that there are provisions provided for default values. This is recommended so that the output is at predictable and useful in the case that the user does not actually change the values through the parameters dialog.

```
Private m_Weight1 As Double = 1.0
Private m_Weight2 As Double = 1.0

Private m_Value1 As Double
Private m_Value2 As Double

Private m_Summary As String = ""
```

Figure 8 – Declaration of model variables

So far we have created the model class, implemented the IDssModel interface, created the parameters page, filled in the implementation of ShowDialog and handled parameters to and from the Model Manager. We still need to handle inputs, outputs, summary reports and the actual execution of the model.

Inputs are stored in the DSSData.mdb database. The Model Manager will provide the inputs you request as long as they exist in the database. The most common place to work with inputs is in the *Execute* function. Figure 9 contains the code in the Execute function of the sample model. Notice that there are several parameters that were not used in this example:

- TimeStep: The TimeStep parameter is used when a loop node is utilized on the layout to execute more than once. This parameter is here for the models that require a time step to run.
- StartDate/StopDate: Some models require a start and stop date as an additional parameter. The user can select a start and stop date in the Model Manager that applies to all models being run.
- ScenarioID: Each scenario is given a unique ID so that outputs from different scenarios can be compared and analyzed side by side.

```
Public Function Execute(ByVal DssManager As DssIntfcLib.IDssManager, _
    ByVal TimeStep As Integer, ByVal DataPath As String, _
    ByVal StartDate As Date, ByVal EndDate As Date, _
    ByVal ScenarioID As String) As Boolean _
    Implements DssIntfcLib.IDssModel.Execute

    Dim input_path As String

    ' Get the input from the DSSData.mdb database
    input_path = DssManager.GetFilePath("SampleModelInputFile")

    If System.IO.File.Exists(input_path) Then
        ' Open the input file and read the data from it
        Dim stream As System.IO.StreamReader

        stream = System.IO.File.OpenText(input_path)
        m_Value1 = System.Convert.ToDouble(stream.ReadLine())
        m_Value2 = System.Convert.ToDouble(stream.ReadLine())
        stream.Close()

        ' Run the model now
        Dim result_value As Double
        result_value = (m_Weight1 * m_Value1 + m_Weight2 * m_Value2) / 2

        m_Summary = "Model run succeeded." & vbCrLf & "  "
        m_Summary &= m_Weight1 & " * " & m_Value1 & " + "
        m_Summary &= m_Weight2 & " * " & m_Value2 & " = "
        m_Summary &= result_value
        Return True
     Else
        ' If the input file was not found then report an error
        DssManager.ReportError(Me, "Could not find the input file '" & _
             input_path & "'.  Model could not execute.")
        Return False
    End If
End Function
```

Figure 9 – Execute function

The only remaining task is to return summary reports so that the Model Manager can combine the summary reports from all the models that were run in a layout.

```
Public ReadOnly Property ResultsSummary() As String _
    Implements DssIntfcLib.IDssModel.ResultsSummary
    Get
        Return m_Summary
    End Get
End Property
```

Figure 10 – Results summary

The model is now complete. If you have understood the process to this point you should be able to create a Model Manager wrapper for any model that you have written.

Technical Documentation: Photo Viewer

Last Revision: 06/15/06

Table of Contents

1. *Table Designs*
   - *Schema, Keys, Indexes and Structure*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*

1. Table Designs

There are two types of Table Designs available for the Photo Viewer. The Table Design is dependent on the type of viewer that is being used. Design #1 is correct if the data being used is organized by Location and then by Photo. Design #2 is correct if the data being used is organized by Watershed, then Stream, then Segment, then Unit, and finally by Photo.

Database: Design #1

Table: Comments
Description: Contains the information needed to select and display the Comments and Comment Information for the selected Photo or Location. Data is selected by either Location_ID or by Location_ID and Photo_ID depending on whether the user is viewing the comments for the location, or for an individual photo.

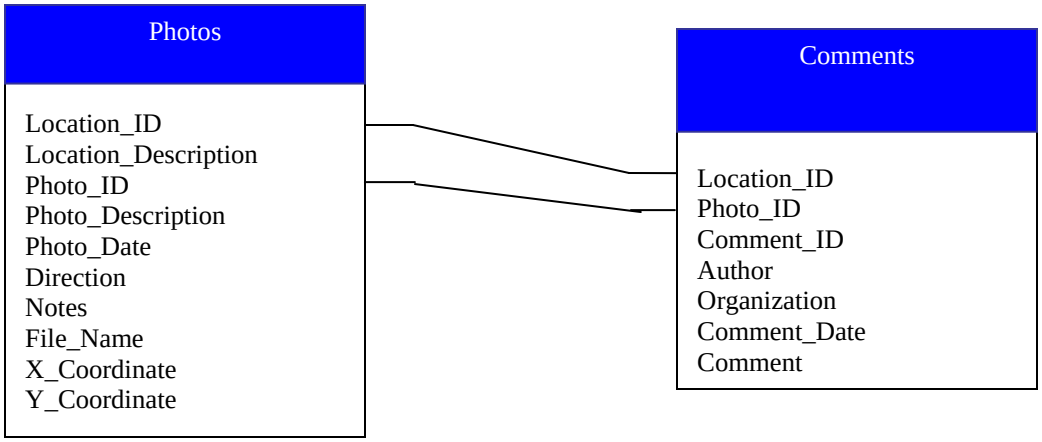
Field	Type	Size	Key Field	Primary Key	Comments
Location_ID	Numeric		No	No	The unique ID for the Location
Photo_ID	Numeric		No	No	The unique ID for the Photo for the given location
Comment_ID	Numeric		No	No	The unique ID for the comment for the given photo
Author	Text	50	No	No	The Author of the comment
Organization	Text	50	No	No	The Organization the Author belongs to
Comment_Date	Text	50	No	No	The date of the comment
Comment	Memo		No	No	The Comment

Table: Photos
Description: Contains the information needed to select and display the Photo and Photo Information for the selected Location. Data is selected by either Location_ID, or by Location_ID, and Photo_ID depending on whether a new photo is being loaded or a new location is being loaded.

Field	Type	Size	Key Field	Primary Key	Comments
Location_ID	Numeric		No	No	The unique ID for the Location
Location_Description	Text	50	No	No	The Location Name
Photo_ID	Number		No	No	The unique ID for the Photo for the given Location

Photo_Description	Text	255	No	No	A description of the photo
Photo_Date	Text	255	No	No	The date the Photo was taken
Direction	Text	50	No	No	The direction of the Photo
Notes	Text	255	No	No	Any Notes about the photo
File_Name	Text	255	No	No	The filename of the Photo
X_Coordinate	Numeric		No	No	X-Coordinate value of the location
Y_Coordinate	Numeric		No	No	Y-Coordinate value of the location

3179



3180

3181

Database: Design #2

Table: Watersheds
Description: Contains the Watersheds for the availablePhotos.

Field	Type	Size	Key Field	Primary Key	Comments
Watershed_ID	Numeric		Yes	Yes	The unique ID for the Watershed
Watershed_Name	Text	255	No	No	The name for the Watershed
Watershed_Description	Text	50	No	No	A description for the Watershed

Table: Streams
Description: Contains the Streams for the Watersheds for the available Photos. Stream_ID's are unique for each Watershed. This means that if there are multiple Watersheds, then there can be duplicate Stream_ID's as long as the Watershed_ID is different.

Field	Type	Size	Key Field	Primary Key	Comments
Watershed_ID	Numeric		Yes	No	The ID for the Watershed that the Stream belongs to
Stream_ID	Numeric		Yes	No	The unique ID for the Stream
Stream_Name	Text	255	No	No	The name for the Stream
Stream_Description	Text	255	No	No	A description for the Stream

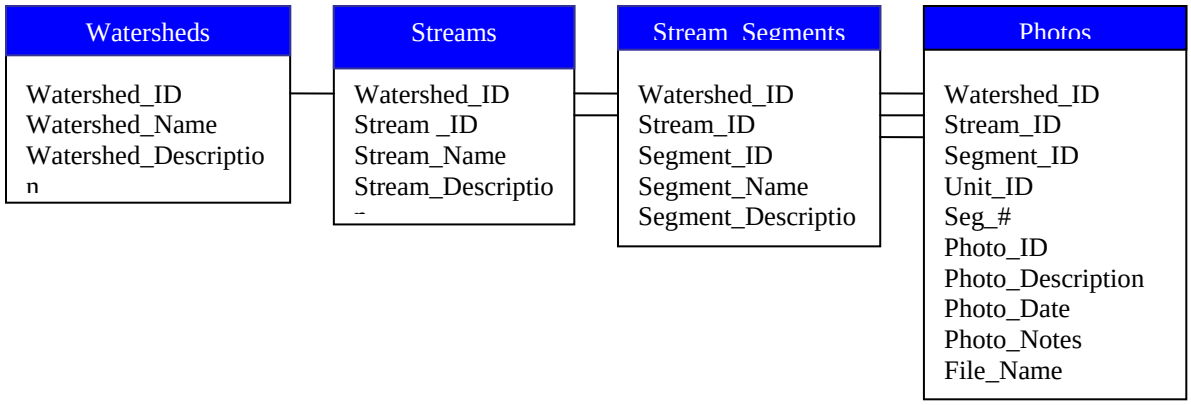
Table: Stream_Segments
Description: Contains the Segments for the Streams for available Photos. Segment_ID's are unique for each Stream. This means that if there are multiple Streams and/or Watersheds, then there can be duplicate Segment_ID's as long as the combination of the Stream_ID and Watershed_ID are different.

Field	Type	Size	Key Field	Primary Key	Comments
Watershed_ID	Numeric		Yes	No	The ID for the Watershed that the Segment belongs to
Stream_ID	Numeric		Yes	No	The ID for the Streams that the Segment belongs to
Segment_ID	Numeric		Yes	No	The unique ID for the Segment
Segment_Name	Text	255	No	No	The name for the Segment
Segment_Description	Text	255	No	No	The description for the Segment

Table: Photos

Description: Contains the information needed to select and display the Photo and Photo Information for the selected Watershed, Stream, Segment, Unit, Photo group. Photo_ID's are unique for each Unit. This means that if there are multiple Units, Segments, Streams, and/or Watersheds, then there can be duplicate Photo_ID's as long as the combination of the Photo_ID, Unit_ID, Segment_ID, Stream_ID and Watershed_ID are different.

Field	Type	Size	Key Field	Primary Key	Comments
Watershed_ID	Numeric		Yes	No	The ID for the Watershed
Stream_ID	Numeric		Yes	No	The ID for the Stream
Segment_ID	Numeric		Yes	No	The ID for the Segment
Unit_ID	Numeric		Yes	No	The ID for the Habitat Unit
Seg_#	Numeric		No	No	The ID for the Habitat Unit within the Segment
Photo_ID	Numeric		Yes	No	The unique ID for the Photo
Photo_Description	Text	255	No	No	The description for the Photo
Photo_Date	Date/Time		No	No	The date the Photo was taken
Photo_Notes	Text	255	No	No	Any notes about the Photo
File_name	Text	255	No	No	The Filename for the Photo



2. Data Needs

There are three types of data needed for the Photo Viewer plug-in. It needs a point shapefile, a folder location, and a database.

- Point Shapefile – this needs to be in the same projection as the other shapefiles. It contains the point locations of where the photos were taken. This file will have a tag that will be associated with it by the plug-in or project file.
- Folder Location – this is the location of all of the images that will be viewed using the Photo Viewer plug-in.
- Database – this needs to be organized according to one of the database designs described in *Section 1: Table Design*.

The layer corresponding with the point shapefile stated above is marked with the tag "PhotoViewerShpfl". If the user removes the layer associated with tag without first closing the Photo Viewer plug-in, then the settings for the PhotoViewer will be reset, and if the Project is then saved, then the settings in the project file will be over-written. Depending on the type of Database structure, the shapefile requires certain fields to be available. Below are the field names and how they correlate to the database.

Required data layers and fields used from each:

1. Design #1 – the following field names need to be named this way exactly.
   a. LOC_ID – corresponds with Location_ID in the database
   b. LOC_NAME – corresponds with Location_Description in the database

2. Design #2 – the following field names do not need to be named this way
   a. Watershed_Name – corresponds with Watershed_Description in the Watersheds table in the database
   b. Stream_Name – corresponds with Stream_Description in the Streams table in the database
   c. Segment_Name – corresponds with Segment_Description in the Stream_Segments table in the database
   d. Unit_ID – corresponds with the Unit_ID in the Photos table in the database
   e. Photo_ID – corresponds with the Photo_ID in the Photos table in the database
   f. X_Coordinate – used for plotting the location on the map in the viewer
   g. Y_Coordinate – used for plotting the location on the map in the viewer

3. Dependencies

The PhotoViewer plug-in requires the following software components and modules to be installed:

Software:
MapWindow 3.1
Visual Studio .NET 2002

Modules and Components:
MapWinGIS Map Control
MapWindow Interfaces

Integrated Development Environment (IDE)
Visual Studio .NET 2002

4. Setup

To begin using the Photo Viewer, it must first be loaded into MapWindow. Once it is loaded, to use the Photo Viewer simply click the camera icon, , on the toolbar. If the data associations (as described in *Section 2: Data Needs*) have been properly formed, then the Photo Viewer will appear and will be fully functional.

If the data associations have not been formed, then a Connection Form will appear allowing you to set the shapefile and associated field(s), database, and photo location folder. Then, after the data associations have been formed, when the Photo Viewer icon, , on the Map Window toolbar is clicked, the viewer will appear and can then be used.

5. Code Compiling

Compiling the Photo Viewer is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 3: Dependencies* are present, load the project into Visual Studio .Net 2002. This Plugin was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
camera.ico	Camera icon for the forms and representing where photos are on the shapefile.

	NOTE: this file needs to be an embedded resource
camera.bmp	Camera bitmap used in the MapWindow Legend for the associated layer. *NOTE: this file needs to be an embedded resource*
clsMainMWI.vb	Contains a class that implements the MapWindow plugin interface.
FolderUtils.vb	Contains a class to access and browse for a folder location.
frmCommentAdder.vb	Displays the Comment form that is used to add comments to a photo or a location if using Database Design #1 (see *Section 1: Table Design* for a description).
frmCommentViewer.vb	Displays the Comments available from the database for the current location or photo if using Database Design #1 (see *Section 1: Table Design* for a description).
frmDBConnection.vb	Displays the Connection form so the user can select the type of data view, the shapefile, photo location, and database.
frmLocationAdder.vb	Displays the form that is used to add a new location to the shapefile through the Photo Viewer.
frmPhotoAdder.vb	Displays the form to add a photo to the current location.
frmPhotoViewer.vb	Displays the Photo Viewer form for Database Design #1 (see *Section 1: Table Design* for a description).
frmPluginKey.vb	A form to enter a Plug-in Key so that the user can specify or edit the connection settings. *NOTE: This form is no longer being shown, but is being left in the project.*
frmSelectLayer.vb	Displays the form to select the associated point shapefile with the Photo Viewer plug-in. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk.
frmViewer2.vb	Displays the Photo Viewer form for Database Design #2 (see *Section 1: Table Designs* for a description).
globalFunctions.vb	This module contains functions used throughout the project for accessing the database, searching shapefiles, initialization, access folder information, and other important functionality.
globalStructs.vb	This module contains the definition of data Structs used throughout the Photo Viewer.
globalVariables.vb	This module contains the variables used throughout the forms, such as the form declaration variables, MapWindow variables, and others.
ImageConverter.vb	Contains a class that implements functions that allows you to convert images to and from an IPictureDisp object. This is needed so that the VBCompatibility.dll does not need to be referenced. This class allows the Photo Viewer image, , to be associated with the points on the associate shapefile.
luginInfo.vb	Contains a class that implements an interface to access (read from and write to) the Project File.

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2002, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

AxMapWinGIS	CopyLocal = True
MapWinGIS	CopyLocal = True
MapWinInterfaces	CopyLocal = False
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Design	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwPhotoViewer.dll has now been created with Microsoft Visual Studio .NET 2002. It is created in the *mwPhotoViewer* subdirectory in the *\Plugins* folder. Next time that MapWindow is run, if the mwPhotoViewer.dll was built to the correct folder, the updated changes to Photo Viewer will be available.

Technical Documentation: Periodicity Viewer/Editor

Last Revision: 06/15/06

Table of Contents

1. *Table Design*
   - *Schema, Keys, Indexes and Structure*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*
   - *Project Files*
   - *Reference Settings*

1. Table Design

Following is a list of tables that must be included with the Periodicity Viewer/Editor Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters. This database must be an Access (*.mdb) database.

Table: tblDistribution_Master
Description: Contains the Master copy of the Fish Distribution Data for each Node, Fish pair. When the default Fish Distribution values are restored, they are restored from this table of values.

Field	Type	Size	Key Field	Primary Key	Comments
DM_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each record in the tblDistribution_Master table.
Node_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Node that this set of Fish Distribution values belong to. It corresponds with the Node_ID Field from the tblNodes table and the tblPeriodicity table.
Lifestage_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the Lifestage ID for the Fish that this set of Fish Distribution values belong to. It corresponds with the Lifestage_ID Field from the tblLifestages table and the tblPeriodicity table.
Current_Known	Yes/No		No	No	This is the "Current Known" Fish Distribution Set value for the given node, fish pair. *NOTE: Yes = Present, No = Absent*
Current_Presumed	Yes/No		No	No	This is the "Current Presumed" Fish Distribution Set value for the given node, fish pair. *NOTE: Yes = Present, No = Absent*
Presumed_Historic	Yes/No		No	No	This is the "Presumed Historic/Potential" Fish Distribution Set value for the given node, fish pair. *NOTE: Yes = Present, No = Absent*

Table: tblLifestages
Description: Contains the Lifestage Information for each of the Species that are available.

Field	Type	Size	Key Field	Primary Key	Comments
Lifestage_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	Unique ID for each Lifestage available. It corresponds with the Lifestage_ID Field in the tblDistribution_Master table, the tblPeriodicity table, and the tblPeriodicity_Master table.
Species_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Species that this Lifestage belongs to. It corresponds with the Species_ID Field in the tblSpecies table.
Lifestage_Name	Text	255	No	No	Name of the Lifestage.

Table: tblNodes
Description: Contains the Node Information (Node ID, Drainage Info, Reach Info) for all available nodes.

Field	Type	Size	Key Field	Primary Key	Comments
pk_NodeID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Node that has Periodicity and/or Fish Distribution available.
Node_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the NodeID that is used to reference the Node. It is the value that corresponds with NodeID that stored in the associated Point Shapefile. It also corresponds with the Node_ID Field in the tblDistribution_Master table, the tblPeriodicity table, and the tbl_Periodicity_Master table.
Drainage_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Drainage that this node belongs to.
Drainage_Name	Text	255	No	No	This is the Name of the Drainage that this node belongs to.
Reach_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Reach that this node belongs to.

Reach_Name	Text	255	No	No	This is the Name of the Reach that this node belongs to.

Table: tblPeriodicity
Description: Contains the current Periodicity and Fish Distribution values set by the user. These are the values that are displayed by the Periodicity Viewer/Editor. *NOTE: The values are initialized to the Master data sets (values found in the tblDistribution_Master table and the tblPeriodicity_Master table).*

Field	Type	Size	Key Field	Primary Key	Comments
Periodicity_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of Periodicity and Fish Distribution values in the table.
Node_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Node that this set of Periodicity and Fish Distribution values belong to. It corresponds with the Node_ID Field in the tblDistribution_Master table, the tblNodes table, and the tblPeriodicity_Master table.
Lifestage_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Fish that this set of Periodicity and Fish Distribution values belong to. It corresponds with the Lifestage_ID Field in the tblDistribution_Master table, the tblLifestages table, and the tblPeriodicity_Master table.
Jan_1	Number	Integer	No	No	Periodicity Value for the 1st half of January for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jan_2	Number	Integer	No	No	Periodicity Value for the 2nd half of January for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Feb_1	Number	Integer	No	No	Periodicity Value for the 1st half of February for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*

Feb_2	Number	Integer	No	No	Periodicity Value for the 2nd half of February for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Mar_1	Number	Integer	No	No	Periodicity Value for the 1st half of March for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Mar_2	Number	Integer	No	No	Periodicity Value for the 2nd half of March for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Apr_1	Number	Integer	No	No	Periodicity Value for the 1st half of April for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Apr_2	Number	Integer	No	No	Periodicity Value for the 2nd half of April for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
May_1	Number	Integer	No	No	Periodicity Value for the 1st half of May for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
May_2	Number	Integer	No	No	Periodicity Value for the 2nd half of May for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jun_1	Number	Integer	No	No	Periodicity Value for the 1st half of June for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jun_2	Number	Integer	No	No	Periodicity Value for the 2nd half of June for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jul_1	Number	Integer	No	No	Periodicity Value for the 1st half of July for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Jul_2	Number	Integer	No	No	Periodicity Value for the 2nd half of July for the given fish, node pair.

					Note: 0=Not Present, 1=Present, 2=Peak
Aug_1	Number	Integer	No	No	Periodicity Value for the 1st half of August for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Aug_2	Number	Integer	No	No	Periodicity Value for the 2nd half of August for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Sep_1	Number	Integer	No	No	Periodicity Value for the 1st half of September for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Sep_2	Number	Integer	No	No	Periodicity Value for the 2nd half of September for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Oct_1	Number	Integer	No	No	Periodicity Value for the 1st half of October for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Oct_2	Number	Integer	No	No	Periodicity Value for the 2nd half of October for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Nov_1	Number	Integer	No	No	Periodicity Value for the 1st half of November for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Nov_2	Number	Integer	No	No	Periodicity Value for the 2nd half of November for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Dec_1	Number	Integer	No	No	Periodicity Value for the 1st half of December for the given fish, node pair.

					Note: 0=Not Present, 1=Present, 2=Peak
Dec_2	Number	Integer	No	No	Periodicity Value for the 2nd half of December for the given fish, node pair. *Note: 0=Not Present, 1=Present, 2=Peak*
Current_Known	Yes/No		No	No	This is the "Current Known" Fish Distribution value for the given node, fish pair. *NOTE: Yes = Present, No = Absent*
Current_Presumed	Yes/No		No	No	This is the "Current Presumed" Fish Distribution value for the given node, fish pair. *NOTE: Yes = Present, No = Absent*
Presumed_Historic	Yes/No		No	No	This is the "Presumed Historic/Potential" Fish Distribution value for the given node, fish pair. *NOTE: Yes = Present, No = Absent*

Table: tblPeriodicity_Master

Description: Contains the Master copy of the Periodicity Data for each Fish pair. When the default Periodicity values are restored, they are restored from this table of values.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
PM_ID	AutoNumber	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each set of Master Periodicity values in the table.
Lifestage_ID	Number	Long Integer	Yes (Duplicates OK)	No	This is the ID for the Fish that this set of Periodicity values belong to. It corresponds with the Lifestage_ID Field in the tblLifestages table and the tblPeriodicity_Master table.
Jan_1	Number	Integer	No	No	Periodicity Value for the 1st half of January for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Jan_2	Number	Integer	No	No	Periodicity Value for the 2nd half of January for the given fish. Note: 0=Not Present, 1=Present, 2=Peak

Feb_1	Number	Integer	No	No	Periodicity Value for the 1st half of February for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Feb_2	Number	Integer	No	No	Periodicity Value for the 2nd half of February for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Mar_1	Number	Integer	No	No	Periodicity Value for the 1st half of March for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Mar_2	Number	Integer	No	No	Periodicity Value for the 2nd half of March for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Apr_1	Number	Integer	No	No	Periodicity Value for the 1st half of April for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Apr_2	Number	Integer	No	No	Periodicity Value for the 2nd half of April for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
May_1	Number	Integer	No	No	Periodicity Value for the 1st half of May for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
May_2	Number	Integer	No	No	Periodicity Value for the 2nd half of May for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Jun_1	Number	Integer	No	No	Periodicity Value for the 1st half of June for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Jun_2	Number	Integer	No	No	Periodicity Value for the 2nd half of June for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Jul_1	Number	Integer	No	No	Periodicity Value for the 1st half of July for the given fish. Note: 0=Not Present, 1=Present,

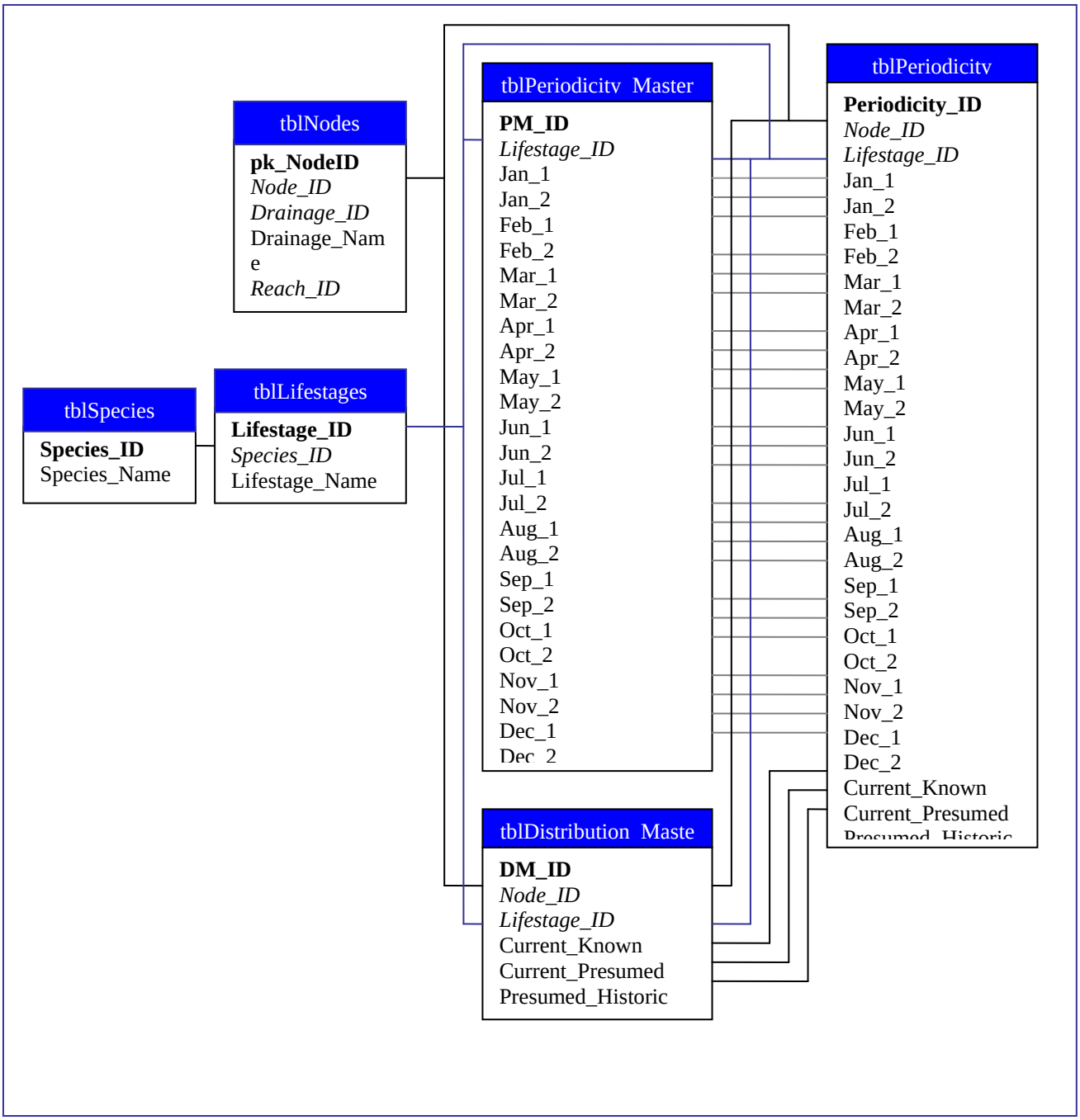
					2=Peak
Jul_2	Number	Integer	No	No	Periodicity Value for the 2nd half of July for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Aug_1	Number	Integer	No	No	Periodicity Value for the 1st half of August for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Aug_2	Number	Integer	No	No	Periodicity Value for the 2nd half of August for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Sep_1	Number	Integer	No	No	Periodicity Value for the 1st half of September for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Sep_2	Number	Integer	No	No	Periodicity Value for the 2nd half of September for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Oct_1	Number	Integer	No	No	Periodicity Value for the 1st half of October for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Oct_2	Number	Integer	No	No	Periodicity Value for the 2nd half of October for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Nov_1	Number	Integer	No	No	Periodicity Value for the 1st half of November for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Nov_2	Number	Integer	No	No	Periodicity Value for the 2nd half of November for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Dec_1	Number	Integer	No	No	Periodicity Value for the 1st half of December for the given fish. Note: 0=Not Present, 1=Present, 2=Peak
Dec_2	Number	Integer	No	No	Periodicity Value for the 2nd half of

					December for the given fish. Note: 0=Not Present, 1=Present, 2=Peak

Table: tblSpecies

Description: Contains the Species Information (ID, Name) for each one available.

Field	Type	Size	Key Field	Primary Key	Comments
Species_ID	Auto Number	Long Integer	Yes (No Duplicates)	Yes	This is a unique ID for each Species. It corresponds with the Species_ID Field in the tblLifestages table.
Species_Name	Text	255	No	No	This is the Name of the Species.



2. Data Needs

There are two type of data needed for the Periodicity Viewer/Editor: a point shapefile, and an Access Database

- Point Shapefile – this needs to be in the same projection as the other shapefiles in your project. It contains the point locations of where each of the Nodes are located. There needs to be field in this shapefile containing the Node_ID values for each of the nodes.

- Access Database – this is the database of Periodicity and Fish Distribution data. This database needs to be organized as described in *Section1: Table Design.*

As stated above, the shapefile requires one field to be found in the data table. Following is the description of the required Field in the shapefile (an example field name, the field type, a description of the data that gets entered into this field, and how it is associated with the database).

Required data layer field description:

a. Node_ID – Integer – The Node ID for the node at this location. It corresponds with the Node_ID field in the tblNodes table in the database. Each value that is found in this field (in the shapefile) also needs to be found in the database.

3. Dependencies

The Periodicity Viewer/Editor plug-in requires the following software components and modules to be installed:

Software:
MapWindow 3.1
Visual Studio .NET 2003

Modules and Components:
MapWindow Interfaces
MapWinGIS Map Control
Microsoft FlexGrid Control 6.0 (SP3)

Integrated Development Environment (IDE)
Visual Studio .NET 2003

4. Setup

To begin using the Periodicity Viewer/Editor, it must first be loaded into MapWindow. Once it is loaded, click on the periodicity icon, , located on the Map Window toolbar If the shapefile and database associations (as described in *Section 2: Data Needs*) have been configured properly and the database is accessible, then the Periodicity Viewer/Editor will appear allowing the user to select and view Periodicity and Fish Distribution data for available nodes.

If the data associations have not been properly set, then a Connection Form will appear allowing you to set the shapefile and database. Then, after the shapefile and database have been properly associated, select the periodicity icon, , from the Map Window toolbar and the viewer will appear allowing the user to view the data.

See the User's Manual for more information and details on associating the data with the Periodicity Viewer/Editor.

5. Code Compiling

Compiling the Periodicity Viewer/Editor is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 3: Dependencies* are present, load the project into Visual Studio .Net 2003. This Plugin was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsPeriodicityMain.vb	Contains a class that implements the MapWindow plugin interface.
frmDBConnection.vb, frmDBConnection.resx	Displays the Connection form that allows the user to associate the shapefile, shapefile field, and database with the Periodicity Viewer/Editor plug-in.
frmEditDistribution.vb, frmEditDistribution.resx	Displays the form that allows the user to Edit the selected Fish Distribution value.
frmEditPeriodicity.vb, frmEditPeriodicity.resx	Displays the form that allows the user to Edit the selected Periodicity value.
frmPeriodicity.vb, frmPeriodicity.resx	Displays the Periodicity Viewer/Editor form. This form contains the table that displays the Periodicity and Fish Distribution data contained in the associated database. You show this form by selecting the periodicity icon, , from the Map Window toolbar.
frmRDESelect.vb, frmRDESelect.resx	Displays the form that allows the user to select which types of data (Periodicity and/or Fish Distribution) to either Restore Defaults for, or to Export (save, print, or copy).
frmSelectLayer.vb, frmSelectLayer.resx	Displays the form to select the point shapefile associated with the Periodicity Viewer/Editor plug-in. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk.
ImageConverter.vb	Contains a class that implements functions that allows you to convert images to and from an IPictureDisp object. This is needed so that the VBCompatibility.dll does not need to be referenced. This class allows the periodicity image, , to be associated with the points on the associated shapefile.
modColorDefinitions.vb	This module contains the color definitions used in the table for displaying the Periodicity Values on the main viewer form.
modDBDefinitions.vb	This module contains the variable definitions that hold the table and field variable names for the Periodicity database. These variables are used throughout the project so that if anything should change in the database, the value only has to be changed in 1 location.

modGlobals.vb	This module contains the variables used throughout the forms, such as the form declaration variables, MapWindow variables, and others.
modPDDefinitions.vb	This module contains the variable definitions that hold the Periodicity and Fish Distribution labels and values for displaying, exporting, and restoring the data.
modUtils.vb	This module contains functions that are used throughout the project for accessing the database, reporting errors, conversions, file functionality, and other necessary functionality.
Periodicity3.bmp	Bitmap version of the Periodicity3.ico. It is used as a custom image for the point shapefile associated with the Periodicity Viewer/Editor plug-in. *NOTE: this file needs to be an embedded resource*
Periodicity3.ico	Icon that is used as the Map Window Legend picture when using a custom image for the point shapefile associated with the Periodicity Viewer/Editor. It also is the icon on the Map Window Toolbar for the Periodicity Viewer/Editor plug-in. *NOTE: this file needs to be an embedded resource*
PluginInfo.vb	Contains a class that implements an interface to access (read from and write to) the Project File.

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

AxMapWinGIS	CopyLocal = True
AxMSFlexGridLib	CopyLocal = True
MapWinGIS	CopyLocal = True
MapWinInterfaces	CopyLocal = False
MSFlexGridLib	CopyLocal = True
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwPeriodicityViewer.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwPeriodicityViewer* subdirectory in the *\Plugins* folder. Next time that MapWindow is run, if the mwPeriodicityViewer.dll was built to the correct folder, the updated changes to the Periodicity Viewer/Editor will be available.

Technical Documentation: Surface Water Quantity Model

Last Revision: Dec 28, 2007

Table of Contents

1. *Overview*
   - *TOPNET Fortran Executable*
   - *DSS Interface*
2. *Interface Table Designs*
   - *Table Descriptions and Notes*
3. *Data*
4. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*
5. *Setup*
   - *Setup*
   - *Customizations*
6. *Building*

1. Overview

The Water Quantity Model is comprised of two components (a) the TOPNET Fortran Executable that performs the model simulations and calculations, and (b) the Water Quantity Model Interface that configures the input to this through the decision support system. The TOPNET Model is described in the Task 4.1 report "Surface Water Quantity Model Development and Calibration". The electronic appendix to this report includes descriptions of the file formats used by this model. This documentation describes the databases and information used by the Water Quantity Model Interface as a plugin component to the DSS.

2. Table Designs

Land Cover Database: LandCoverSummary.mdb
(See “Technical Documentation For Land Cover Summarizer” for table designs)

Water Management Databases (For each Water Quantity Model dataset: Historic, Existing, and Full Build Out): HistoricWaterManagement.mdb, ExistingWaterManagement.mdb, FBOWaterManagement.mdb

Table: MonthlyDemand. Provides information on monthly variation of water demand.

Field Name	Data Type	Description
InYearDemandType	Long Integer	Index matching the InYearDemandType field in User table.
Name	Text	Monthly Demand Name
Month1	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month2	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month3	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month4	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month5	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month6	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.

Month7	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month8	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month9	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month10	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month11	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.
Month12	Long Integer	Multipliers to adjust the daily demand for this month with respect to default daily demand rate.

Table: Reservoir. Provides data on reservoirs to water management model.

Field Name	Data Type	Description
ReservoirID	Long Integer	Unique identifier for reservoir
Name	Text	Reservoir Name
DrainageID	Long Integer	WRIA1 drainage where reservoir is located.
InOffStream	Long Integer	Reservoir type: 1 - Instream, 2 - Offstream
RightID	Long Integer	ID of water right for this reservoir
MaxStore	Double	Maximum storage (m^3)
InitialStore	Double	Initial storage (m^3)
MinStore	Double	Minimum storage (m^3)
MaxInFlow	Double	Maximum inflow (m^3/s)
MaxWithdrawal	Double	Maximum withdrawal (m^3/s)
MinEnvRelease	Double	Minimum environment release (m^3/s)
LossRate	Double	Loss rate (m^3/s)

Table: ReturnFlow. Provides information on return flows for the water management model.

Field Name	Data Type	Description
ReturnFlowID	Long Integer	Unique identifier for return flow matching ReturnFlowID in User table.
Name	Text	Return Flow Name

NumReturnFlows	Long Integer	Number of return flows associated with user.
ReturnFlowUnits	Long Integer	ReturnFlowUnits: 1=FracFlow, 2=FracMinDemand, 3=Volume
ReturnFlowAmt1	Double	Amount of first return flow, fraction or volume. Units are m^3/day if ReturnFlowUnits=3, volume..
ReturnFlowType1	Long Integer	Type of return flow: 1 - Surface Water, 2 - Ground Water, 3 - Reservoir.
ReturnFlowLocn1	Long Integer	WRIA1 drainage ID of drainage where first return flow occurs. Specifying 0 means the drainage associated with the user producing this return flow.
ReturnFlowWWTPID1	Long Integer	Identifier of wastewater treatment plant (WWTP) if return flow is treated.
ReturnFlowAmt2	Double	Amount of second return flow, fraction or volume. Units are m^3/day if ReturnFlowUnits=3, volume..
ReturnFlowType2	Long Integer	Type of return flow: 1 - Surface Water, 2 - Ground Water, 3 - Reservoir.
ReturnFlowLocn2	Long Integer	WRIA1 drainage ID of drainage where second return flow occurs. Specifying 0 means the drainage associated with the user producing this return flow.
ReturnFlowWWTPID2	Long Integer	Identifier of wastewater treatment plant (WWTP) if return flow is treated.

Table: Rights. Provides information on water rights to water management model.

Field Name	Data Type	Description
RightID	Long Integer	Water Rights Identifier matches RightID in User table.
Name	Text	Name of Water Right
PriorityDate	Date/Time	Priority date.
LegalDailyMax	Double	Maximum daily withdrawal in cubic meters per day.
LegalAnnMax	Double	Maximum annual withdrawal in cubic meters.

Table: SeasonsDefn. Provides start and end day for seasons.

Field Name	Data Type	Description
SeasonsDefnID	Long Integer	Unique identifier for a season definition record
Name	Text	Name of Season Definition
StartDaySeason1	Long Integer	Day to start season 1
StartDaySeason2	Long Integer	Day to start season 2
StartDaySeason3	Long Integer	Day to start season 3
StartDaySeason4	Long Integer	Day to start season 4

Table: Source. Provides information on user sources to the water management model.

Field Name	Data Type	Description
SourceID	Long Integer	Unique ID Number
Name	Text	Name of source
Type	Long Integer	1 - Surface Water, 2 - Ground Water, 3 – Reservoir
SourceLocationID	Long Integer	Drainage ID or Reservoir ID
PhysicalDailyMax	Double	Physical Limitation (m^3/day)
PhysicalAnnMax	Double	Physical Limitation (m^3/day)

Table: SourceMixing. Provides information about the apportioning of take between the sources that the users draw water from.

Field Name	Data Type	Description
SourceMixingID	Long Integer	Source mixing index from User table. There may be multiple records with the same index as there is a one (record in User) to many (records in sourcemixing) relationship.
Name	Text	Name of source mixing record
UsersSourceNum	Long Integer	Matches the UsersSourceNum in UserSourceRight table
Units	Long Integer	1 - fraction, 2 - volume
Season1Amount	Double	How much to take from this source for season 1 in the season defined by SeasonsDefnID. If volume the units are m^3/day.
Season2Amount	Double	How much to take from this source for season 2 in the season defined by SeasonsDefnID. If volume the units are m^3/day.
Season3Amount	Double	How much to take from this source for season 3 in the season defined by SeasonsDefnID. If

		volume the units are m^3/day.
Season4Amount	Double	How much to take from this source for season 4 in the season defined by SeasonsDefnID. If volume the units are m^3/day.
SeasonsDefnID	Long Integer	ID of row in SeasonsDefinition Table

Table: SourceMixingOld. Earlier format of SourceMixing table. Not used unless importing information from water management files.

Field Name	Data Type	Description
SourceMixingID	Long Integer	Source mixing index from User table. There may be multiple records with the same index as there is a one (record in User) to many (records in sourcemixing) relationship.
Name	Text	Name of source mixing
UsersSourceNum	Long Integer	Matches the source ID in User table
Units	Long Integer	1 - fraction, 2 - volume
Amount	Double	How much to take from this source. If volume the units are m^3/day.
SeasonNumber	Long Integer	Number from 1 to 4
SeasonsDefnID	Long Integer	ID of row in SeasonsDefinition Table

Table: User. Provides information about water users and water demands to the water management component.

Field Name	Data Type	Description
UserID	Long Integer	Unique sequential id for user
Name	Text	Name of user
UserType	Long Integer	User type in one of the following categories: 1 - SoilMoistureIrrigation 2 - FixedDemandIrrigation 3 - DownstreamReservoirRelease 4 - PWS 5 - NonPWSMandI 6 - Dairy 7 - Ranch 8 - Poultry 9 - ParkGolfCemetery 10 - InstreamFlow 11 - Diversion 12 - ReservoirFill 13 - InStreamReservoirRelease 14 - OffStreamReservoirRelease

POU_ID	Long Integer	Place of Use. The WRIA1 Drainage Identifier DrainID where the use occurs.
DemandVble	Double	Number of quantifiable user units (e.g. people or cows). Set to one if not relevant. (The demand is calculated by multiplying this variable with the DemandRate.)
DemandRate	Double	The amount of water demand per unit time for one unit of activity (m^3/day/unit)
InYearDemandType	Long Integer	Index to record in Monthly Demand Table that specifies the monthly (seasonal) variation associated with this use. InYearDemandType may be set to zero if the use is constant over the year or to -1 if this information is not required, such as is the case for irrigation users where demand is based on soil moisture and reference evaporation.
ReturnFlowID	Long Integer	Index to record in return flow table that specifies information about the return flow associated with this user.
SourceMixingID	Long Integer	Index to records in source mixing table in file sourcemixing.txt that specifies information about apportioning of take between the sources that the user draws water from. This is a one to many relationship.

Table: UserSourceRight. Table that associates sources and rights with users

Field Name	Data Type	Description
UserID	Long Integer	User id of user associated with a source and water right
SourceID	Long Integer	Id of source used by specified user
RightID	Long Integer	Id of water right associated with the specified user
UsersSourceNum	Long Integer	Id of record in Source SourceMixingTable giving the proportioning between sources associated with a user

Water Quantity Simulated Streamflow Database: WaterQuantityStreamFlow.mdb and WaterQuantityStreamFlow.mdf

(See “Technical Documentation For Time Series Analyst” for table designs)

3. Data

The Water Quantity Model requires the following input data:

- Drainages Shapefile
- TopNet Input Directory
- Land Cover Summary Database
- Water Management Database

The drainage shapefile used should be bsnwria1_v7.shp. Only the drainage identifier (BSNWRIA1A) and drainage name (DRAIN_NAME) are used from this shapefile. Other inputs have been created using the drainage identifiers in this shapefile. Use of another shapefile would result in incorrect names being associated with drainage identifiers. The WRIA-1 drainages shapefile is installed at the following location by default:

C:\Program Files\WRIA-1_DSS\DSS_Data\GIS_Data\Shapefiles\Watershed\bsnwria1_v7.shp

The TopNet input directory is the folder comprising input files required to run the water quantity model. Three such folders are provided, one each for historic, existing and full build out conditions. When the model is run this entire set of files is copied to the specified output working directory. The model is then run in the output working directory which serves as a record of the inputs and outputs for a specific model run scenario. The TopNet Input Directories are installed at the following locations by default:

C:\Program Files\WRIA-1_DSS\Water Quantity TopNet Files\ModelInputFilesHistoric
C:\Program Files\WRIA-1_DSS\Water Quantity TopNet Files\ModelInputFilesExisting
C:\Program Files\WRIA-1_DSS\Water Quantity TopNet Files\ModelInputFilesFBO

The land cover summary database contains the area of each land cover type summarized for each drainage. This is created by the land cover summarizer tool. The initial land cover summary database provided includes land covers for each scenario:

- Scenario 1 = Historic conditions
- Scenario 2 = Existing conditions
- Scenario 3 = Full build out conditions

The Land Cover Summary Database is installed at the following location by default:

C:\Program Files\WRIA-1_DSS\DSS_Data\Model Manager Databases\LandCoverSummary.mdb

The water management database is a Microsoft Access database contains the information needed to populate the water quantity model water management input files. This database is changed when water management options are edited. Three initial water management databases are provided, one for each scenario. The Water Management Databases are installed at the following locations by default:

C:\Program Files\WRIA-1_DSS\DSS_Data\Model Manager Databases\HistoricWaterManagement.mdb

C:\Program Files\WRIA-1_DSS\DSS_Data\Model Manager Databases\ExistingWaterManagement.mdb
C:\Program Files\WRIA-1_DSS\DSS_Data\Model Manager Databases\FBOWaterManagement.mdb

The Water Quantity Model produces a large number of text output files. These are saved in a folder created each time the model is run. By default these folders are created in:

C:\Program Files\WRIA-1_DSS\Model Runs.

The formats of these output files are described in the electronic appendix to the Task 4.1 report "Surface Water Quantity Model Development and Calibration". The modeled Streamflow from this model output, from the file FlowAtStreamNodes_cms.txt is added to the WaterQuantityStreamflow database designated in the output tag of the Water Quantity Model Interface. This database is in the format used by the Time Series Analyst Component and documented in the Technical Documentation for Time Series Analyst. The Water Quantity Interface model adds data to the WQData table and a new Parameter record identifying the scenario simulated to the Parameter_CodeDescriptions table. All other tables are left unchanged.

Three output databases are provided at the following locations by default:

C:\Program Files\WRIA-1_DSS\MSDEDATABASES\WaterQuantityStreamFlow.mdf
C:\Program Files\WRIA-1_DSS\MSDEDATABASES\WaterQuantityStreamFlowSimRef.mdf
C:\Program Files\WRIA-1_DSS\DSS_Data\Model Manager Databases\WaterQuantityStreamflow.mdb

The first and third of these databases have no streamflow data in them, but are pre-loaded with the station information corresponding to the nodes at which streamflow is simulated. The second of these WaterQuantityStreamflowSimRef.mdf contains four reference simulations (Historic, Existing without water management, Existing and Full Build out) that USU conducted for the 45 year period of record (10/1/1959 to 12/31/2005).

The following GIS files are provided with the DSS to facilitate interpretation of the Surface Water Quantity Model output:

- points_of_interest_v8.shp
- Nodes.shp
- Nooknet.shp

points_of_interest_v8.shp identifies the points where streamflow is simulated by the TOPNET model, including names and station identifiers used by WRIA. Nodes.shp provides a cross reference between the sequential node numbering that TOPNET uses internally and the node numbering on points_of_interest_v8.shp. See the Task 4.1 report "Surface Water Quantity Model Development and Calibration" for the node numbering schema definitions. Nooknet.shp is the stream network shapefile representing stream reaches derived during the process of setting

up the Water Quantity Model and depicting the connectivity between nodes used in the simulation. By default these shapefiles are installed in:

C:\program files\WRIA-1_DSS\DSS_Data\GIS_Data\Shapefiles\Watershed

4. Dependencies

The Water Quantity Model Interface requires the following software to be installed:

Software:

MapWindow 3.1

DSS Model Manager

Integrated Development Environment (IDE):

Visual Studio .NET 2003 Complete Install

The Water Quantity Model requires the data output by the following DSS elements:

Land Cover Summary Tool

The Water Quantity Model Fortran Executable is provided as a standalone executable without any dependencies.

5. Setup

Water Quantity Model Interface

The WRIA-1_DSS Installation installs the Water Quantity Model in the MapWindow Plugins folder.

(This is usually located at C:\Program Files\MapWindow\Plugins\ModelManager\Elements").

6. Building

To compile the Water Quantity Model Interface Management Practices Tool, add the files below to a Microsoft Visual Studio .NET 2003 Visual Basic .NET Library Project.

mwWaterQuantityModel project files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
BasinParsFile.vb	Contains the routines used to write an updated basinpars.txt file for the TopNet model input.
DBClient.vb	Contains the routines used to connect to a database.
DirectoryPicker.vb	Custom component that allows a user to select a directory from the computer's file system.

EditMonthlyDemandData.vb	Custom component allowing users to edit monthly demand data for water management purposes.
EditReservoirData.vb	Custom component allowing users to edit reservoir data for water management purposes.
EditReturnFlowData.vb	Custom component allowing users to edit return flow data for water management purposes.
EditRightsData.vb	Custom component allowing users to edit water rights data for water management purposes.
EditSeasonsDefinitionsData.vb	Custom component allowing users to edit seasons definition data for water management purposes.
EditSourceData.vb	Custom component allowing users to edit source data for water management purposes.
EditUserData.vb	Custom component allowing users to edit user data for water management purposes.
EditWatershedInfo.vb	Custom component allowing users to view watershed information for water management purposes.
FlowWriter.vb	Contains the routines used to write flow data output by the TopNet model to a simulated streamflow database.
Folder.ico	An icon used in frmWaterRights to represent a group of users within a WRIA-1 Drainage.
frmArtificialDrainages.vb	Allows the user to view/edit the artificial drainage (a.k.a. agricultural drainage) values for a WRIA-1 Drainage.
frmConfiguration.vb	Contains the routines used to select a database.
frmEditExistingItem.vb	Allows the user to select an existing water management item to edit.
frmEditImperviousArea.vb	Allows the user to edit impervious area data for a WRIA-1 drainage.
frmEditNewItem.vb	Allows the user to select a new water management time to edit.
frmEditSourceMixing.vb	Allows the user to select the source mixing for a water user’s souces.
frmEditSourceMixingPeriods.vb	Allows the user to edit a source mixing period.
frmImperviousAreas.vb	Allows the user to edit the impervious area data for a WRIA-1 drainage.
frmInputs.vb	Allows the user to select the data locations for the model inputs and outputs.
frmProgress.vb	A form containing a progress bar to show the progress of an activity.
frmSelectLayer.vb	Contains the routines used to select a layer from MapWindow’s view.
frmSelectModelDirectory.vb	A form containing a DirectoryPicker component which allows the user to select a directory from the file system, or a directory to be created by the Water Quantity Model in the current DSS Scenario Run.
frmSelectScenario.vb	Allows the user to select a previously run DSS scenario by id.

frmSelectSources.vb	Allows the user to select water sources for a water user.
frmWaterRights.vb	Allows the user to edit water management for the Water Quantity Model.
ImperviousAreas.xml	Contains the default impervious areas and artificial drainage values. Written to disk next to mwWaterQuantityModel.dll if it doesn't exist.
ImperviousXMLFile.vb	Contains the methods used to manage the impervious area data xml file.
LUParsFIle.vb	Contains the methods used to read default basin parameters from the file lupars.csv.
NewWaterRightsItem.vb	Allows the user to create a new water rights item.
OpenFolder.ico	An icon used in frmWaterRights to represent a group of users within a WRIA-1 Drainage which is currently displayed.
Reservoir.ico	An icon used in frmWaterRights to represent a reservoir.
Source.ico	An icon used in frmWaterRights to represent a water source.
TopInpFile.vb	Contains the routines used to write an updated topinp.dat file for the TopNet model input.
WaterMgmtControlFile.vb	Contains the routines used to write an updated WatermgmtControl.txt file for the TopNet model input.
WaterQuantityModel.vb	Implements MapWindow Plugin and DSS Interface routines, allowing the Lake Whatcom Model to act as a MapWindow Plugin and a DSS element.
WaterRights.xml	Contains lists of water management types used in frmWaterRights. Written to disk next to mwWaterQuantityModel.dll if it doesn't exist.
WaterRightsWriter.vb	Contains the methods to manage the water management files for the Water Quantity Model.
WaterUser.ico	An icon used in frmWaterRights to represent the root water user node.
WaterUserClosedFolder.ico	An icon used in frmWaterRights to represent a water user that is not currently selected.
WaterUserOpenFolder.ico	An icon used in frmWaterRights to represent a water user that is currently selected.

Add the following references to the project:

- DssIntfcLib.dll
- ICSharpCode.SharpZipLib.dll
- MapWinGIS.ocx
- MapWinInterfaces.dll
- System.dll
- System.Data.dll
- System.Drawing.dll
- System.Windows.Forms.dll
- System.Xml.dll

You are now ready to compile the project by clicking the Build Solution menu option under the menu Build.

Water Quantity Model Fortran Executable

The executable topnet.exe is included in each input folder and replicated in each output folder. Topnet.exe is compiled using the Compaq Visual Fortran version 6.6 compiler from the following files.

Source Code

AllocateWaterToUsers.f90	hydatasn_v7.f	read_inputs.f90
Append_To_Output_Tables.f90	ImposeMeasuredFlows.f90	read_struct_from_text.f90
AssignDrainageFlows.f90	indepth1_v7.f	snowdgtv22.f
AssignPriorityOrder.f90	Initialise_Output_Tables.f90	SnowLSub.f
BalanceFlowsAtReservoirs.f90	kinwav_v7.f	SnowWrap.f
BalanceFlowsAtStreamNodes.f90	Luns.inc	snowxv22.f
BuildDrainageOrder.f90	maxvariables.inc	Tdims_v7.inc
BuildLinkStructure.f90	mddata_v10_7.f	toplim_v7.f
BuildNodeStructure.f90	MIC.FOR	topmoddgt_v7.f
CalculateDemand.f90	modules.f90	top_main_v7.f
calv46sn_v7.f	newt.for	types.f90
Data_Read.f90	Nlfit.inc	watermgmt.f90
et_v7_fromdave.f	PropagateWaterViaUser.f90	Write_Output_line.f90
find.f90	PTRATE.FOR	Write_Output_Tables.f90

Compaq Visual Fortran Project Files

topnet.dsw
topnet.ncb
topnet.opt
compilable\compilable.dsp
compilable\compilable.plg

These source code and project files are provided in the electronic appendix to the Task 4.1 report "Surface Water Quantity Model Development and Calibration"

The Water Quantity Model is installed in all versions of the WRIA-1 DSS installation.

Technical Documentation:
Temperature and DO Flags Model

Table of Contents

1. *Stream Response Model Output Folder Data Needs*

2. *Watershed Loading Model Output Text File Data Needs*

3. *Input Model Nodes Shapefile Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*
   - *Necessary Fields*
   - *Selecting Nodes for Modeling*

4. *Output Temperature and DO Database Table Design*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

5. *Output Temperature Flags and Warnings Shapefile Data Needs*
   - *Type/Location of Data*
   - *Necessary Fields*

6. *Output DO Flags and Warnings Shapefile Data Needs*
   - *Type/Location of Data*
   - *Necessary Fields*

7. *Dependencies*
   - *Software*
   - *Modules and Components*
   - *Integrated Development Environment (IDE)*

8. *Setup*
   - *Setup*

9. *Code Compiling*
   - *Project Files*
   - *Reference Settings*

1. Stream Response Model Output Folder Data Needs

This is the folder containing the output text files from the Waterbody Response Model. The files in this folder must be named in the following way:

NodeTS###.txt

Where ### represents the three digit node number for that file. If the files in this folder are not named in this way and are not the Waterbody Response Model output files, the Temperature and DO Flags Model will not run.

2. Watershed Loading Model Output Text File Data Needs

This text file is the output from the Watershed Loading Model. It must be a text file. It can have any name.

3. Input Model Nodes Shapefile Data Needs

There are two types of data required for the Input Model Nodes Shapefile: a point shapefile path, and a *Node ID* shapefile field.

- Point Shapefile – this shapefile needs to be in the same project as the other shapefiles in your project. It contains the point locations of each of the Model Nodes. Connection Name – this is the name of the Time Series Analyst Connection.
- *Station* Shapefile Field – this field is a String field that contains the Node ID for each of the Model Nodes. It corresponds with the *Station* Field in the Stations table and with the *Station* Field in the *WQData* table for the Output Temperature DO Database.
  NOTE: If the Node ID in the shapefile is not found in either of the Input Databases, then the model will not run,and there will be an entry in the log explaining why.

In order to specify the Nodes to model, the Model Nodes Shapefile must be selected and added to the project. Then you can either select the which Nodes to model by specifying the Node ID's or by going to map and selecting with the Map Window Selection tools.

4. Output Temperature and DO Database Table Design

Following is a list of tables that must be included with the Output Temperature and DO Database. Other tables and information may exist, and not all the tables listed are used by the Temperature and DO Flags Model. However, these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters.

Table: Agency_Code Descriptions
Description: This table is not used by the Temperature and DO Flags Model.

Field	Type	Size	Key Field	Primary Key	Comments
Agency_ID	Number	Long Integer	No	No	
Agency_Code	Text	50	Yes (No Duplicates)	Yes	
Agency_Description	Text	255	No	No	
Contact	Text	50	No	No	

Table: Comment_Code Descriptions
Description: This table is not used by the Temperature and DO Flags Model and is blank.

Field	Type	Size	Key Field	Primary Key	Comments
Comment_Code	Text	50	Yes (No Duplicates)	Yes	
Description	Text	200	No	No	

Table: Data_Code Descriptions
Description: This table is not used by the Temperature and DO Flags Model and is blank.

Field	Type	Size	Key Field	Primary Key	Comments
Data_Code	Text	50	Yes (No Duplicates)	Yes	
Description	Text	255	No	No	

Table: GroupNames
Description: This table is not used by the Temperature and DO Flags Model and is blank.

Field	Type	Size	Key Field	Primary Key	Comments
Group_ID	Number	Long Integer	Yes (Duplicates OK)	No	
Group_Name	Text	50	No	No	

Table: GroupStations

Field	Type	Size	Key Field	Primary Key	Comments
Key	AutoNumber	Long Integer	Yes (Duplicates Ok)	No	
Group_ID	Number	Long Integer	Yes (Duplicates OK)	No	The ID for the Group. It corresponds with the *Group_ID* Field in the *GroupNames* table.
Station	Text	50	No	No	

Table: Parameter_Code Descriptions

Field	Type	Size	Key Field	Primary Key	Comments
Parameter_Code	Number	Long Integer (Automatic decimal places)	Yes (No Duplicates)	Yes	A unique ID for each Parameter.
Parameter_Name	Text	255	No	No	
Class	Text	50	No	No	
Common_Name	Text	50	No	No	

Table: QAQC_Code Descriptions
Description: This table is not used by the Temperature and DO Flags Model and is blank.

Field	Type	Size	Key Field	Primary Key	Comments
QAQC_Code	Text	50	Yes (No Duplicates)	Yes	
Description	Text	150	No	No	

Table: Source_Database Descriptions

Field	Type	Size	Key Field	Primary Key	Comments
Source_Database	Text	50	Yes (No Duplicates)	Yes	
Description	Text	255	No	No	
Source_Contact	Text	50	No	No	

Table: Stations

Description: This table contains Station data for all Stations available to model.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	50	Yes (No Duplicates)	Yes	
Shapefile	Number	Long Integer	No	No	
USU_Subbasin	Text	50	No	No	
Station_Name	Text	100	No	No	
Location	Text	200	No	No	
State	Text	50	No	No	
County	Text	50	No	No	
Latitude	Number	Double (#.#####)	No	No	
Longitude	Number	Double (#.#####)	No	No	
UTMX	Number	Double (#.##)	No	No	
UTMY	Number	Double(#.##)	No	No	
River_Mile	Number	Double	No	No	
Station_Type	Text	75	No	No	
NHDID	Text	50	Yes (Duplicates OK)	No	
Wb_Name	Text	50	No	No	
WRIA1_Shed	Text	50	No	No	

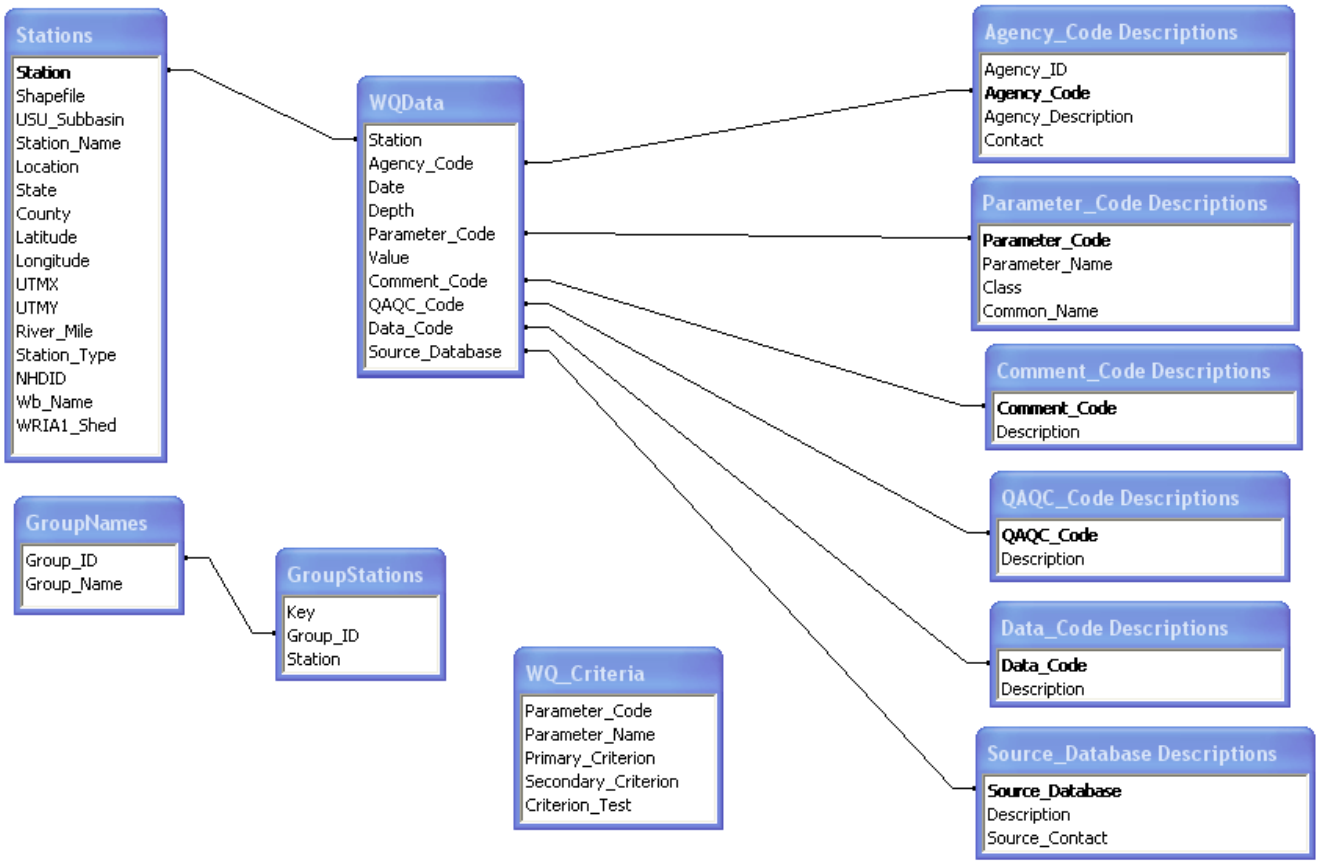
Table: WQData

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	255	Yes (Duplicates OK)	No	This corresponds with the *Station* Field in the Stations Table
Agency_Code	Text	255	Yes (Duplicates OK)	No	This corresponds with the *Agency_Code* Field in the Agency_Code Descriptions Table
Date	Date/ Tme	mm/dd/ yyyy	No	No	

		hh:nn AM/PM			
Depth	Number	Double	No	No	
Parameter_Code	Number	Long Integer	Yes (Duplicates OK)	No	This corresponds with the *Parameter_Code* Field in the Parameter_Code Descriptions Table
Value	Number	Double	No	No	
Comment_Code	Text	255	Yes (Duplicates OK)	No	This corresponds with the *Comment_Code* Field in the Comment_Code Descriptions Table
QAQC_Code	Text	255	Yes (Duplicates OK)	No	This corresponds with the *QAQC_Code* Field in the QAQC_Code Descriptions Table
Data_Code	Text	255	Yes (Duplicates OK)	No	This corresponds with the *Data_Code* Field in the Data_Code Descriptions Table
Source_Database	Text	255	Yes (Duplicates OK)	No	This corresponds with the *Source_Database* Field in the Source_Database Descriptions Table

Table: WQ_Criteria
Description: This table is not used by the Temperature and DO Flags Model and is blank.

Field	**Type**	**Size**	**Key Field**	**Primary Key**	**Comments**
Parameter_Code	Number	Long Integer	Yes (Duplicates Ok)	No	
Parameter_Name	Text	50	No	No	
Primary_Criterion	Number	Double	No	No	
Secondary_Criterion	Number	Double	No	No	
Criterion_Test	Text	50	No	No	


Stations
Station
Shapefile
USU_Subbasin
Station_Name
Location
State
County
Latitude
Longitude
UTMX
UTMY
River_Mile
Station_Type
NHDID
Wb_Name
WRIA1_Shed
WQData
Station
Agency_Code
Date
Depth
Parameter_Code
Value
Comment_Code
QAQC_Code
Data_Code
Source_Database
Agency_Code Descriptions
Agency_ID
Agency_Code
Agency_Description
Contact
Parameter_Code Descriptions
Parameter_Code
Parameter_Name
Class
Common_Name
Comment_Code Descriptions
Comment_Code
Description
QAQC_Code Descriptions
QAQC_Code
Description
Data_Code Descriptions
Data_Code
Description
Source_Database Descriptions
Source_Database
Description
Source_Contact
GroupNames
Group_ID
Group_Name
GroupStations
Key
Group_ID
Station
WQ_Criteria
Parameter_Code
Parameter_Name
Primary_Criterion
Secondary_Criterion
Criterion_Test

5. Output Temperature Flags and Warnings Shapefile Data Needs

The Temperature Flags and Warnings Shapefile must be the shapefile named Temp_Flags_Warnings.shp. Data is written to this shapefile whenever the Temperature and DO Flags Model is run. Any existing data is erased from the database and the new model data is written. This shapefile must contain the following fields:

- *Shape* – All data in this shapefile are Points
- *Station* – This field contains the Station name for each Node. This Field corresponds with the *Station* Shapefile Field in the Input Model Nodes Shapefile and the *Station* Field in the Stations table and with the *Station* Field in the *WQData* table for the Output Temperature DO Database
- *Station_na* – This fields contains the Drainage information for each Node.
- *UTMX*- This is the X value for the UTM coordinates for each Node.
- *UTMY*- This is the Y value for the UTM coordinates for each Node.
- *Site_type*- This specifies the node type, for example, if the node is a USU Intensive site or a Mixing Node, etc.
- *Num_flags*- This field contains the number of Temperature Flags at each Node.
- *Num_warn*- This field contains the number of Temperature Warnings at each Node.
- *Color*- If the *Num_flags* Field has a value greater than zero, the color is Red. If the *Num_flags* Field has a value of zero and the *Num_warn* Field has a value greater than zero, the color is Yellow. If both *Num_flags* and *Num_warn* Fields contain a value of zero AND the *Station_na* Field does not say "No WQ Data", the color is Green. If the *Station_na* Field says "No WQ Data", the color is Black.

6. Output DO Flags and Warnings Shapefile Data Needs

The DO Flags and Warnings Shapefile must be the shapefile named DO_Flags_Warnings.shp. Data is written to this shapefile whenever the Temperature and DO Flags Model is run. Any existing data is erased from the database and the new model data is written. This shapefile must contain the following fields:

- *Shape* – All data in this shapefile are Points
- *Station* – This field contains the Station name for each Node. This Field corresponds with the *Station* Shapefile Field in the Input Model Nodes Shapefile and the *Station* Field in the Stations table and with the *Station* Field in the *WQData* table for the Output Temperature DO Database
- *Station_na* – This fields contains the Drainage information for each Node.
- *UTMX*- This is the X value for the UTM coordinates for each Node.
- *UTMY*- This is the Y value for the UTM coordinates for each Node.
- *Site_type*- This specifies the node type, for example, if the node is a USU Intensive site or a Mixing Node, etc.
- *Num_flags*- This field contains the number of DO Flags at each Node.

- *Num_warn*- This field contains the number of DO Warnings at each Node.
- *Color*- If the *Num_flags* Field has a value greater than zero, the color is Red. If the *Num_flags* Field has a value of zero and the *Num_warn* Field has a value greater than zero, the color is Yellow. If both *Num_flags* and *Num_warn* Fields contain a value of zero AND the *Station_na* Field does not say "No WQ Data", the color is Green. If the *Station_na* Field says "No WQ Data", the color is Black.

7. Dependencies

The Time Series Analyst plug-in requires the following software components and modules to be installed:

Software:
MapWindow 3.1
Visual Studio .NET 2003

Modules and Components:
DSS Interface Definitions
MapWindow Interfaces
MapWinGIS Components

Integrated Development Environment (IDE)
Visual Studio .NET 2003

8. Setup

To begin using the Temperature and DO Flags Model, you must first load the Scenario Builder/DSS Plug-in into Map Window. Once it is loaded, click on the DSS Menu located on the Map Window Menu bar and select either Scenario Builder, or the name of the Scenario that you want to run. Once the Scenario Builder is and the Temperature and DO Flags Model is added to the current Scenario, Run the Scenario. If all the input and output parameters have been correctly set and they are all accessible, then the Temperature and DO Flags Model will run for the selected data. After it is finished the user may view the run log if they desire. If any errors occur during the run, an error log will appear explaining the error.

If the input and output data associations have not been properly set, then double click on the Temperature and DO Flags Model scenario node, and a Parameters Form will appear allowing you to edit the input and output data. Then, after the parameters have been properly associated, Run the Scenario and after it is finished, the user may view the run log if they desire or if any errors occurred, then an error log will appear explaining the error.

See the User's Manual for more information and details on setting the parameters for the Temperature and DO Flags Model.

9. Code Compiling

Compiling the Temperature and DO Flags Model is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 7: Dependencies* are present, load the project into Visual Studio .Net 2003. This DSS Model was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsMainPhabTS.vb	Contains a class that implements the Map Window plug-in interface and the DSS Model Interface and other functions for editing the parameters for and running the Temperature and DO Flags Model.
frmConfiguration.vb, frmConfiguration.resx	A Database Configuration form. This form is shown when the user edits an Input or Output Database. It allows the user to specify the Path for an Access Database. The connection for the specified values can then be tested to make sure a valid database is specified. To save the selected settings, click on the Save Changes button; to cancel the changed settings, close the form with the Red X button at the top of the form. This form will only be shown if the user decides to edit any of the Input or Output databases already associated with the Temperature and DO Flags model.
frmParameters.vb, frmParameters.resx	A form that allows the user to view and edit the set the Input and Output Parameters for the Temperature and DO Flags Model.
frmSelectLayer.vb, frmSelectLayer.resx	A form to select the Input Model Nodes point shapefile for the Temperature and DO Flags Model. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk. This form will only be shown if the user decides to edit the shapefile already associated with the Temperature and DO Flags Model.
frmSelectNodes.vb, frmSelectNodes.resx	A form that allows the user to select the nodes to model, either from the Map Window Map or from the list of available nodes. This form will only be shown if the user decides to edit the selected nodes already associated with the Temperature and DO Flags Model.
modDatabase.vb	Contains functions for access and writing to the Input and Output database, and the variable definitions for the Output Database type.
modGlobals.vb	Contains the definitions for variables used throughout the project, such as MapWindow variables, model variables, parameter variables, error variables, and others.

modParameters.vb	Contains the variable definitions for the Input Parameter Names.
modShapefile.vb	Contains the variable definitions and functions for accessing data on the associated Model Nodes Shapefile.
modUtils.vb	Contains functions used throughout the project for reporting errors, accessing files, and other functionality.

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 7: Dependencies*).

Reference Settings:

ADODB	CopyLocal = False
DssIntfcLib	CopyLocal= True
MapWinGIS	CopyLocal = False
MapWinInterfaces	CopyLocal = False
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwTempDOflags.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwTimeSeries* subdirectory in the

\Plugins\ModelManager\Elements\mwTempDOFlags

folder. Next time that MapWindow is run, if the mwTempDOflags.dll was built to the correct folder, the updated changes to the Temperature and DO Flags Model will be available.

Technical Documentation: Time Series Analyst

Last Revision: 06/15/06

Table of Contents

1. *Table Design*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Modules and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*
   - *Project Files*
   - *Reference Settings*

1. Table Design

Following is a list of tables that must be included with the Time Series Analyst Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters.

Table: Agency_Code Descriptions

Field Name	Data Type	Size	Key Field	Primary Key	Description
Agency_ID	Long Integer		No	No	
Agency_Code	Text	100	No	No	
Agency_Description	Text	255	No	No	
Contact	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

Table: Comment_Code Descriptions

Field Name	Data Type	Size	Key Field	Primary Key	Description
Comment_Code	Text	100	No	No	
Description	Text	400	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

Table: GroupNames

Field Name	Data Type	Size	Key Field	Primary Key	Description
Group_ID	Long Integer		No	No	
Group_Name	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

Table: GroupStations

Field Name	Data Type	Size	Key Field	Primary Key	Description
[Key]	Long Integer		Yes	Yes	
Group_ID	Long Integer		No	No	
Station	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was

					locally modified or locally created.

Table: Parameter_Code Descriptions

Field Name	Data Type	Size	Key Field	Primary Key	Description
Parameter_Code	Long Integer		No	No	
Parameter_Name	Text	100	No	No	
Class	Text	100	No	No	
Common_Name	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

Table: QAQC_Code Descriptions

Field Name	Data Type	Size	Key Field	Primary Key	Description
QAQC_Code	Text	100	No	No	
Description	Text	300	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

Table: Source_Database Descriptions

Field Name	Data Type	Size	Key Field	Primary Key	Description
Source_Database	Text	100	No	No	
Description	Text	100	No	No	
Source_Contact	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

Table: Stations

Field Name	Data Type	Size	Key Field	Primary Key	Description
Station	Text	50	No	No	
Shapefile	Long Integer		No	No	
USU_Subbasin	Text	50	No	No	
Station_Name	Text	50	No	No	
Location	Text	100	No	No	
State	Text	100	No	No	

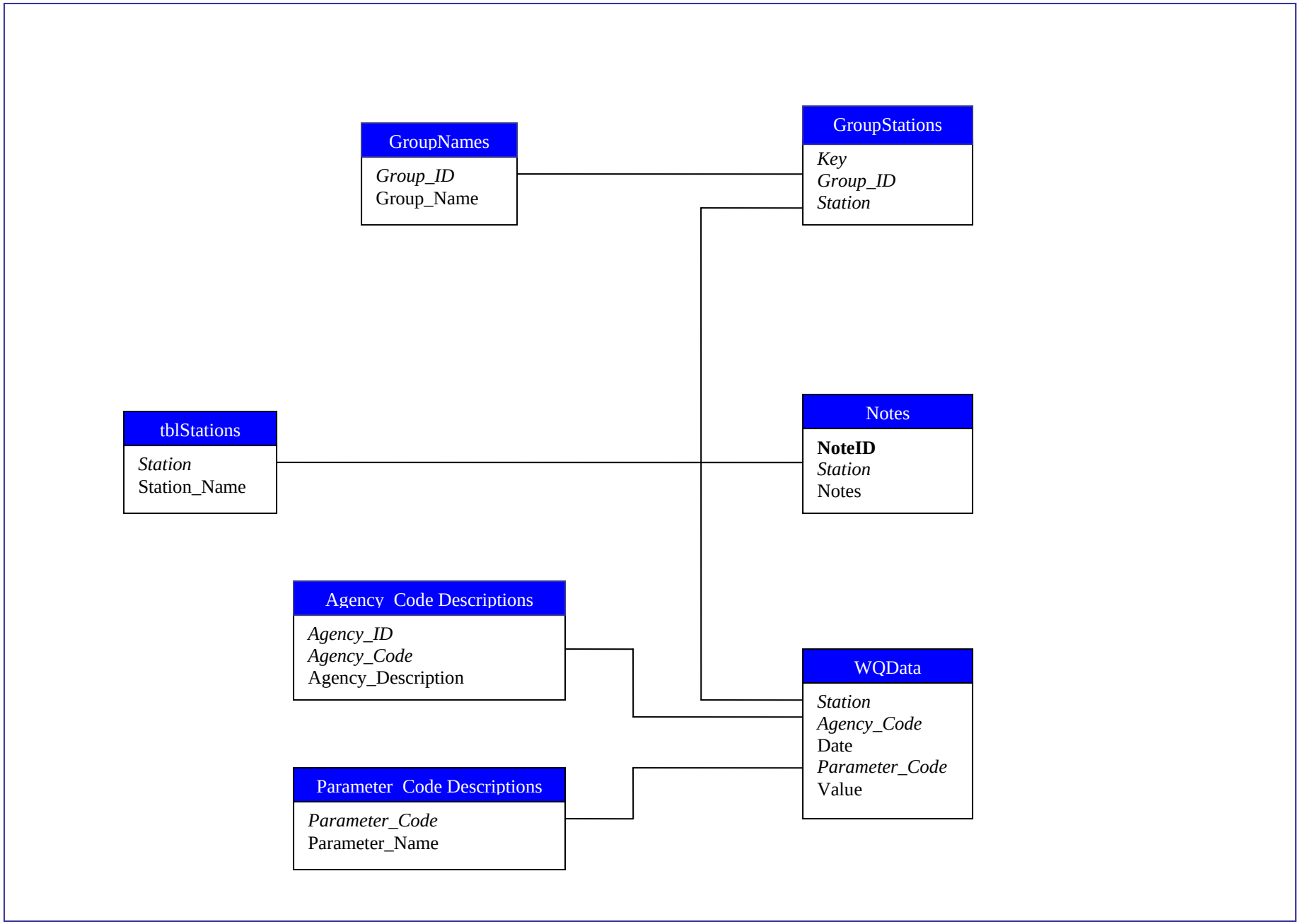
County	Text	100	No	No	
Latitude	Double		No	No	
Longitude	Double		No	No	
UTMX	Double		No	No	
UTMY	Double		No	No	
River_Mile	Double		No	No	
Station_Type	Text	100	No	No	
NHDID	Text	100	No	No	
Wb_Name	Text	100	No	No	
WRIA1_Shed	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

4074
4075
4076 Table: WQ_Criteria

Field Name	Data Type	Size	Key Field	Primary Key	Description
Parameter_Code	Long Integer		No	No	
Parameter_Name	Text	100	No	No	
Primary_Criterion	Double		No	No	
Secondary_Criterion	Double		No	No	
Criterion_Test	Text	100	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.

4077
4078
4079 Table: WQData

Field Name	Data Type	Size	Key Field	Primary Key	Description
Station	Text	255	No	No	
Agency_Code	Text	255	No	No	
Date	Date/Time		No	No	
Depth	Double		No	No	
Parameter_Code	Long Integer		No	No	
Value	Double		No	No	
Comment_Code	Text	255	No	No	
QAQC_Code	Text	255	No	No	
Data_Code	Text	255	No	No	
Source_Database	Text	255	No	No	
LocallyModified	Boolean		No	No	Indicates that data was locally modified or locally created.


GroupNames
Group_ID
Group_Name
GroupStations
Key
Group_ID
Station
tblStations
Station
Station_Name
Notes
NoteID
Station
Notes
Agency Code Descriptions
Agency_ID
Agency_Code
Agency_Description
WQData
Station
Agency_Code
Date
Parameter_Code
Value
Parameter Code Descriptions
Parameter_Code
Parameter_Name

4080

2. Data Needs

There are three types of data required for the Time Series Analyst: a connection name, a database type, and a database. There are also two Optional types of data that can be specified: a point shapefile, and a custom Icon.

Required:

- Connection Name – this is the name of the Time Series Analyst Connection.
- Database Type – this is the type of database that you are connecting to. IE: Access, or SQL Server.
- Database – this is the database of Time Series Data. This database needs to be organized as described in *Section 1. Table Design.*

Optional:

- Point Shapefile – this shapefile needs to be in the same projection as the other shapefiles in your project. It contains the point locations of where each of the Stations are located. There needs to be a field containing the *Station ID* in this shapefile. This file will have a tag that will be associated with it by the plug-in:"WQGAGES|*FieldID*|*ConnectionName*" where *FieldID* is the Field Number for the *StationID* Field in the Shapefile, and *ConnectionName* is the Connection Name described above.
- Custom Icon – this is the icon that you want to be associated with the specified Point Shapefile. Because there can be multiple connections to the Time Series Analyst, you can specify this Icon to be different for each connection created. This can only be specified if you associate a Point Shapefile with a connection.

As stated above, the point shapefile requires one field to be found in the data table. Following is the description of the required Field in the shapefile (an example field name, the field type, a description of the data that gets entered into this field, and how the field data is associated with the data in the database.

Optional Point Shapefile field:

a. *Station ID* – String – The ID for the Station at this location. It corresponds with the *Station* field in the *Stations* table found in the database. Each value that is found in this field (in the shapefile) also needs to be found in the database.

3. Dependencies

The Time Series Analyst plug-in requires the following software components and modules to be installed:

Software:

Gigasoft's ProEssentials Graphing Tools Version 4
MapWindow 3.1
Visual Studio .NET 2003

Modules and Components:
Gigasoft ProEssentials Graph v4
Gigasoft ProEssentials Sci-Graph v4
MapWindow Interfaces
MapWinGIS Components
Microsoft FlexGrid Control 6.0 (SP3)

Integrated Development Environment (IDE)
Visual Studio .NET 2003

4. Setup

To begin using the Time Series Analyst, it must first be loaded into MapWindow. Once it is loaded, click on the Time Series Analyst icon, , located on the Map Window toolbar If the required connection name(s), database type, and database associations, and, if desired, optional shapefile and icon associations (as described in *Section 2: Data Needs*) have been configured properly and the database for the selected connection is accessible, then the Time Series Analyst will appear allowing the user to select and view the data for available stations.

If the data associations have not been properly set, then a Connection Form will appear allowing you to create or edit connections. When you create or edit a connection, a form appears that allows you to specify the required data (the connection name, the database type, and the database), and the optional data (the point shapefile and the custom icon). Then, after the required and optional data have been properly associated, select the Time Series Analyst icon, , from the Map Window toolbar and either the viewer or a form allowing the user to select which connection to view will appear, then the user can select and view the available data.

See the User's Manual for more information and details on associating the data with the Time Series Analyst.

5. Code Compiling

Compiling the Time Series Analyst is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 3: Dependencies* are present, load the project into Visual Studio .Net 2003. This Plugin was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsLine.vb	Contains a class that defines a Line Object.

CommonFunc.vb	Contains variables, classes, enumerations, structs, and functions used in multiple forms and classes throughout the project.
DBInfoLayer.vb	Contains classes for accessing and storing Time Series Analyst connections and their associated settings.
dock.ico	Icon used on several forms to indicate that an Options/Plot Info panel is docked. *NOTE: this file needs to be an embedded resource*
frmAbout.vb, frmAbout.resx	An About Form for the Time Series Analyst.
frmAddCriteria.vb, frmAddCriteria.resx	A form that allows the user to Add or Edit Criteria.
frmAddMTSStations.vb, frmAddMTSStations.resx	A form that allows the user to add or remove Stations or Parameters, depending on the type of plot (*Multiple Station,1 Parameter* or *1 Station, Multiple Parameter*) selected, for the Multiple Time Series plot.
frmAddStations.vb, frmAddStations.resx	A form that allows the user to add or remove Stations to a Group.
frmBoxPlot.vb, frmBoxPlot.resx	A form that explains the Box Plot symbols, shading, and shape.
frmConfiguration.vb, frmConfiguration.resx	An SQL Server Configuration form. This form is shown when the user browses for a Database whose type = SQL. It allows the user to set the Server Address, User ID and Password, and Database Name. The connection for the specified values can then be tested to make sure a valid database is specified.
frmControlLine.vb, frmControlLine.resx	A form that allows the user to add or remove Control Lines for the selected plot.
frmControlLineValues.vb, frmControlLineValues.resx	A form that allows the user to specify or edit the Label, Value, and Color of a selected Control Line.
frmCriteria.vb, frmCriteria.resx	A form that displays the created Criteria and their results.
frmCrossPlot.vb, frmCrossPlot.resx	A form that plots a Correlation and Time Series Plot for two Station, Parameter pairs.
frmCustomQuery.vb frmCustomQuery.resx	A form that allows the user to find the Stations in a selected Time Series Connection that meet certain criteria. The user can search for Stations either by defining Spatial boundaries using the Map, or by defining Database Criteria that must be met.
frmDataBaseConnections.vb, frmDataBaseConnections.resx	A form that allows the user to Add, Edit, or Remove Time Series Analyst Connections. It displays the Connection Name, Database, and Custom Icon (if set) for the created Connections.
frmDBConnection.vb, frmDBConnection.resx	The Connection form that allows the user to associate or edit the required data (the connection name, the database type, and the database) and the optional data (the point shapefile,

	Station ID field, and the custom Icon) for a Time Series Analyst Connection.
frmGroup.vb, frmGroup.resx	The form to create a new Group. It allows the user to specify the new group's Name and ID.
frmHotSpot.vb, frmHotSpot.resx	The form that displays the data for a specific point from any of the plots.
frmListOfCriterias.vb, frmListOfCriterias.resx	A form that displays the list of created Criteria and their settings, and allows the user to add and edit the Criteria.
frmMDBAddMTSStations.vb frmMDBAddMTSStations.resx	A form that allows the user to add or remove Locations or Parameters, depending on the type of plot (*Multiple Locations, 1 Parameter* or *1 Location, Multiple Parameters,* where a location consists of a Database Connection and Station pair) selected, for the Multiple Database, Multiple Time Series plot.
frmMDBCrossPlot.vb, frmMDBCrossPlot.resx	A form that plots a Correlation and Time Series Plot for two Location, Parameter pairs where a Location consists of a Database Connection and Station pair.
frmMDBMultipleTS.vb, frmMDBMultipleTS.resx	A form that plots a Multiple Database, Multiple Time Series Plot for the selected plot type (*Multiple Locations, 1 Parameter* or *1 Location, Multiple Parameters* where a Location consists of a Database Connection and Station pair).
frmMultipleTS.vb, frmMultipleTS.resx	A form that plots a Multiple Time Series Plot for the selected plot type (*Multiple Stations, 1 Parameter* or *1 Station, Multiple Parameters*).
frmNormalizeOptions.vb, frmNormalizeOptions.resx	A form that allows the user to specify settings for Normalizing a plot's values.
frmParameterList.vb, frmParameterList.resx	A form that allows the user to specify which parameters to view on the Parameter Summary plots or table, or to use to create a summary report by parameter for all stations.
frmPluginKey.vb, frmPluginKey.resx	A form to enter a Plug-in Key so that the user can specify or edit Database Connections. *NOTE: This form is no longer being shown, but is being left in the project.*
frmSelectLayer.vb, frmSelectLayer.resx	Displays the form to select the optional point shapefile associated with a Database Connection to the Time Series Analyst plug-in. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk. This form will only be shown if the user decides to associate a shapefile with a Database Connection.
frmSummaryTable.vb, frmSummaryTable.resx	A form that displays a summary of the data in the database by Parameter for a selected Station.
frmWaterQuality.vb frmWaterQuality.resx	Displays the Time Series Analyst form for a selected Database Connection. This form contains the different plots, tables, and summaries for available stations. You show this

	form by selecting the Time Series Analyst icon, , from the Map Window toolbar and then selecting which Database Connection you want to view, or by selecting one or more points on an associated shapefile.
frmWQStations.vb, frmWQStations.resx	A form that displays plots (a *Number Distribution* plot and a *Time Distribution* plot) of the parameter summary for a selected Station.
hide.ico	Icon used on several forms to indicate that an Options/Plot Info panel is hidden. *NOTE: this file needs to be an embedded resource*
ImageConverter.vb	Contains a class that implements functions that allows you to convert images to and from an IPictureDisp object. This is needed so that the VBCompatibility.dll does not need to be referenced. This class allows the Time Series Analyst image, , to be associated with the points on the optional shapefile if it is associated.
Main.vb	Contains a class that implements the MapWindow plugin interface.
PluginInfo.vb	Contains a class that implements an interface to access (read from and write to) the Project File.
Statistics.vb	Contains a class and an enumeration to access, store, and calculate statistics, averaged values, and censored data for the plots in the Time Series Analyst.
TS1.bmp	Bitmap version of the TS1.ico. It is used as the default custom image for the optional point shapefile, when it is associated with a Database Connection in the Time Series Analyst plug-in. *NOTE: this file needs to be an embedded resource*
TS1.ico	Icon that is used as the Map Window Legend picture when using the default image for the optional point shapefile when it is associated with a Database Connection in the Time Series Analyst. It also is the icon on the Map Window Toolbar for the Time Series Analyst plug-in. *NOTE: this file needs to be an embedded resource*

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2003, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

ADODB	CopyLocal = False
AxMSFlexGridLib	CopyLocal = True
AxPEGO32BLib	CopyLocal = True
AxPESGO32BLib	CopyLocal = True
MapWinGIS	CopyLocal = False
MapWinInterfaces	CopyLocal = False

MSFlexGridLib	CopyLocal = True
PEGO32BLib	CopyLocal = True
PESGO32BLib	CopyLocal = True
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwTimeSeries.dll has now been created with Microsoft Visual Studio .NET 2003. It is created in the *mwTimeSeries* subdirectory in the *Plugins* folder. Next time that MapWindow is run, if the mwTimeSeries.dll was built to the correct folder, the updated changes to the Time Series Analyst will be available.

Technical Documentation:
Watershed Characterization Report Generator

Last Revision: 06/25/08

Table of Contents

1. *Table Designs*
   - *Schema, Keys, Indexes and Structure*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*

2. *GIS Data Needs*
   - *Location of GIS Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*
   - *Customizations*

5. *Building*
   - *Code Compiling*
   - *Installations*

1. Table Designs

Database: WRIAReportData

Table: 303d
Description: Contains data regarding 303(d) listed waterbodies. Data is selected by either wcrsNumber, wbid, or wgclNumber, depending on whether it's a waterbody, watercourse, or water grid cell.

Field	Type	Size	Key Field	Primary Key	Comments
LocallyModified	Boolean		No	No	Indicates that data was Locally modified or created.
wcrsNumber	Text	50	No	No	
Parameter	Text	50	No	No	
Medium	Text	50	No	No	
303d	Text	50	No	No	
FirstListedYear	Text	50	No	No	
Basis	Text	2048	No	No	
Remarks	Text	2048	No	No	
ActionNeeded	Text	50	No	No	
wbid	Text	50	No	No	
wgclNumber	Text	50	No	No	

Table: CensusData-CompleteState
Description: Contains census data for the complete state. Data is selected by 'town or city name' field.

Field	Type	Size	Key Field	Primary Key	Comments
town or city name	Text	255	Yes	No	
Type	Text	255	No	No	
Index	Numeric	8	No	No	
Place (FIPS)	Numeric	8	No	No	Federal Information Processing Standard
Internal Point (Latitude)	Numeric	8	No	No	
Internal Point (Longitude)	Numeric	8	No	No	
Total population: Total	Numeric	8	No	No	
Occupied housing units: Average household size; Total	Numeric	8	No	No	

Occupied housing units: Total	Numeric	8	No	No	
Households: Median household income in 1999 ; Total	Numeric	8	No	No	
Total population: Male	Numeric	8	No	No	
Total population: Male; Under 1 year	Numeric	8	No	No	
Total population: Male; 1 year	Numeric	8	No	No	
Total population: Male; 2 years	Numeric	8	No	No	
Total population: Male; 3 years	Numeric	8	No	No	
Total population: Male; 4 years	Numeric	8	No	No	
Total population: Male; 5 years	Numeric	8	No	No	
Total population: Male; 6 years	Numeric	8	No	No	
Total population: Male; 7 years	Numeric	8	No	No	
Total population: Male; 8 years	Numeric	8	No	No	
Total population: Male; 9 years	Numeric	8	No	No	
Total population: Male; 10 years	Numeric	8	No	No	
Total population: Male; 11 years	Numeric	8	No	No	
Total population: Male; 12 years	Numeric	8	No	No	
Total population: Male; 13 years	Numeric	8	No	No	
Total population: Male; 14 years	Numeric	8	No	No	
Total population: Male; 15 years	Numeric	8	No	No	
Total population: Male; 16 years	Numeric	8	No	No	
Total population: Male; 17 years	Numeric	8	No	No	
Total population: Male; 18 years	Numeric	8	No	No	
Total population: Male; 19 years	Numeric	8	No	No	
Total population: Male; 20 years	Numeric	8	No	No	
Total population: Male; 21 years	Numeric	8	No	No	
Total population: Male; 22 to 24 years	Numeric	8	No	No	
Total population: Male; 25 to 29 years	Numeric	8	No	No	
Total population: Male; 30 to 34 years	Numeric	8	No	No	
Total population: Male; 35 to 39 years	Numeric	8	No	No	
Total population: Male; 40 to 44 years	Numeric	8	No	No	

Total population: Male; 45 to 49 years	Numeric	8	No	No	
Total population: Male; 50 to 54 years	Numeric	8	No	No	
Total population: Male; 55 to 59 years	Numeric	8	No	No	
Total population: Male; 60 and 61 years	Numeric	8	No	No	
Total population: Male; 62 to 64 years	Numeric	8	No	No	
Total population: Male; 65 and 66 years	Numeric	8	No	No	
Total population: Male; 67 to 69 years	Numeric	8	No	No	
Total population: Male; 70 to 74 years	Numeric	8	No	No	
Total population: Male; 75 to 79 years	Numeric	8	No	No	
Total population: Male; 80 to 84 years	Numeric	8	No	No	
Total population: Male; 85 years and over	Numeric	8	No	No	
Total population: Female	Numeric	8	No	No	
Total population: Female; Under 1 year	Numeric	8	No	No	
Total population: Female; 1 year	Numeric	8	No	No	
Total population: Female; 2 years	Numeric	8	No	No	
Total population: Female; 3 years	Numeric	8	No	No	
Total population: Female; 4 years	Numeric	8	No	No	
Total population: Female; 5 years	Numeric	8	No	No	
Total population: Female; 6 years	Numeric	8	No	No	
Total population: Female; 7 years	Numeric	8	No	No	
Total population: Female; 8 years	Numeric	8	No	No	
Total population: Female; 9 years	Numeric	8	No	No	
Total population: Female; 10 years	Numeric	8	No	No	
Total population: Female; 11 years	Numeric	8	No	No	
Total population: Female; 12 years	Numeric	8	No	No	
Total population: Female; 13 years	Numeric	8	No	No	

Total population: Female; 14 years	Numeric	8	No	No	
Total population: Female; 15 years	Numeric	8	No	No	
Total population: Female; 16 years	Numeric	8	No	No	
Total population: Female; 17 years	Numeric	8	No	No	
Total population: Female; 18 years	Numeric	8	No	No	
Total population: Female; 19 years	Numeric	8	No	No	
Total population: Female; 20 years	Numeric	8	No	No	
Total population: Female; 21 years	Numeric	8	No	No	
Total population: Female; 22 to 24 years	Numeric	8	No	No	
Total population: Female; 25 to 29 years	Numeric	8	No	No	
Total population: Female; 30 to 34 years	Numeric	8	No	No	
Total population: Female; 35 to 39 years	Numeric	8	No	No	
Total population: Female; 40 to 44 years	Numeric	8	No	No	
Total population: Female; 45 to 49 years	Numeric	8	No	No	
Total population: Female; 50 to 54 years	Numeric	8	No	No	
Total population: Female; 55 to 59 years	Numeric	8	No	No	
Total population: Female; 60 and 61 years	Numeric	8	No	No	
Total population: Female; 62 to 64 years	Numeric	8	No	No	
Total population: Female; 65 and 66 years	Numeric	8	No	No	
Total population: Female; 67 to 69 years	Numeric	8	No	No	
Total population: Female; 70 to 74 years	Numeric	8	No	No	
Total population: Female; 75 to 79 years	Numeric	8	No	No	
Total population: Female; 80 to 84 years	Numeric	8	No	No	
Total population: Female; 85 years	Numeric	8	No	No	

and over					
Employed civilian population 16 years and over: Total	Numeric	8	No	No	
Employed civilian population 16 years and over: Male	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Agricultur	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Agricultu1	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Mining	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Constructi	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Manufactur	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Wholesale	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Retail tra	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Transporta	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Transport1	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Utilities	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Informatio	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Finance; i	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Finance an	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Real estat	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Profession	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Professio1	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Management	Numeric	8	No	No	
Employed civilian population 16	Numeric	8	No	No	

years and over: Male; Administra					
Employed civilian population 16 years and over: Male; Educationa	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Education1	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Health car	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Arts; ente	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Arts; ent1	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Accommodat	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Other serv	Numeric	8	No	No	
Employed civilian population 16 years and over: Male; Public adm	Numeric	8	No	No	
Employed civilian population 16 years and over: Female	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Agricult	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Agricul1	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Mining	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Construc	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Manufact	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Wholesal	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Retail t	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Transpor	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Transpo1	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Utilitie	Numeric	8	No	No	
Employed civilian population 16	Numeric	8	No	No	

years and over: Female; Informat					
Employed civilian population 16 years and over: Female; Finance;	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Finance	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Real est	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Professi	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Profess1	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Manageme	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Administ	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Educatio	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Educati1	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Health c	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Arts; en	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Arts; e1	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Accommod	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Other se	Numeric	8	No	No	
Employed civilian population 16 years and over: Female; Public a	Numeric	8	No	No	
year	Text	50	No	No	
annualPopGrowth	Numeric	8	No	No	
recordnumber	Numeric	4	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.

4276
4277
4278

Table: CensusData-wshed
Description: This table is a way to link the CensusData-completeState table above with drainages in the watershed. PlaceName indicates the place in CensusData-completeState that falls geographically in the drainage specified by DrainName.

Field	Type	Size	Key Field	Primary Key	Comments
PlaceName	Text	50	Yes	No	
DrainName	Text	50	Yes	No	
PlaceIndex	Numeric	8	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: DataSources
Description: sectionName is the name of the heading as shown in the report. DataSource is the datasource text field displayed below the heading. sectionLink corresponds to the sectionLink number in the Crystal Reports themselves; sectionLink is used to link this table to the reports.

Field	Type	Size	Key Field	Primary Key	Comments
sectionName	Text	150	No	No	
DataSource	Text	150	No	No	
sectionLink	Numeric	75	Yes	No	
recordnumber	Numeric	4	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Disclaimer
Description: Contains the USU Disclaimer text displayed near the front of the report. Table contains only one row of data.

Field	Type	Size	Key Field	Primary Key	Comments
Disclaimer	Text	2147483647	No	No	
LocallyModified	Boolean	1	No	No	Local mod or create

Table: Discussion
Description: Legacy Table for adding watershed specific ‘discussion’ comments. Not used by the existing watershed characterization program.

Field	Type	Size	Key Field	Primary Key	Comments
Watershed	Text	50	Yes	No	Watershed Name

Discussion	Text	16	No	No	New Discussion
LocallyModified	Boolean	1	No	No	Data locally modified or created.

Table: Drainages
Description: Contains data regarding drainages within the watershed. All data in the report is selected from drainage name, making it a very important field and a candidate key.

Field	Type	Size	Key Field	Primary Key	Comments
Area	Numeric	8	No	No	Area (m^2)
BSNSWRIA1_ID	Numeric	8	No	No	WRIA 1 Unique ID
WtrUse_ID	Numeric	8	No	No	
DrainName	Text	255	Yes	No	Drainage Name
Acres	Numeric	8	No	No	Acreage
SqMiles	Numeric	8	No	No	
Description	Text	255	No	No	
ID	Numeric	8	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.
hasCoastline	Boolean	1	No	No	

Table: FishDist
Description: Contains fish distribution data by species. For each of the species names given by the field names, the value is 0, 1 or 2. 0 indicates no distribution, 1 indicates distribution, and 2 indicates critical distribution. Drain_name is the drainage in which distribution is concerning.

Field	Type	Size	Key Field	Primary Key	Comments
CHIN	Numeric	8	No	No	
CHUM	Numeric	8	No	No	
PINK	Numeric	8	No	No	
COHO	Numeric	8	No	No	
SOCK	Numeric	8	No	No	
STHD	Numeric	8	No	No	
CUTT	Numeric	8	No	No	
CHAR	Numeric	8	No	No	
KOK	Numeric	8	No	No	
RBT	Numeric	8	No	No	
EBT	Numeric	8	No	No	

FACHIN	Numeric	8	No	No	
SPCHIN	Numeric	8	No	No	
SUSTHD	Numeric	8	No	No	
WSTHD	Numeric	8	No	No	
DRAIN_NAME	Text	255	Yes	No	Drainage Name
recordnumber	Numeric	4	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: FishPeriodicityKC
Description: Fish Periodicity/Presence data for each month, where 0 is no presence, 1 is present, and 2 is critical presence. Data in this table is specifically for the species in the Known Current dataset.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	75	Yes	No	Drainage Name
Species	Text	50	Yes	No	
Lifestage	Text	50	Yes	No	
PresenceJAN	Numeric	2	No	No	
PresenceFEB	Numeric	2	No	No	
PresenceMAR	Numeric	2	No	No	
PresenceAPR	Numeric	2	No	No	
PresenceMAY	Numeric	2	No	No	
PresenceJUN	Numeric	2	No	No	
PresenceJUL	Numeric	2	No	No	
PresenceAUG	Numeric	2	No	No	
PresenceSEP	Numeric	2	No	No	
PresenceOCT	Numeric	2	No	No	
PresenceNOV	Numeric	2	No	No	
PresenceDEC	Numeric	2	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
Recordnumber	Numeric	4	No	No	
presentInDrainage	Boolean	1	No	No	
segment	Text	50	Yes	No	

Table: FishPeriodicityKH

Description: Fish Periodicity/Presence data for each month, where 0 is no presence, 1 is present, and 2 is critical presence. Data in this table is specifically for the species in the Known Historic dataset.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	75	Yes	No	
Species	Text	50	Yes	No	
Lifestage	Text	50	Yes	No	
PresenceJAN	Numeric	2	No	No	
PresenceFEB	Numeric	2	No	No	
PresenceMAR	Numeric	2	No	No	
PresenceAPR	Numeric	2	No	No	
PresenceMAY	Numeric	2	No	No	
PresenceJUN	Numeric	2	No	No	
PresenceJUL	Numeric	2	No	No	
PresenceAUG	Numeric	2	No	No	
PresenceSEP	Numeric	2	No	No	
PresenceOCT	Numeric	2	No	No	
PresenceNOV	Numeric	2	No	No	
PresenceDEC	Numeric	2	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
Recordnumber	Numeric	4	No	No	
presentInDrainage	Boolean	1	No	No	
segment	Text	50	Yes	No	

Table: FishPeriodicityPC

Description: Fish Periodicity/Presence data for each month, where 0 is no presence, 1 is present, and 2 is critical presence. Data in this table is specifically for the species in the Presumed Current dataset.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	75	Yes	No	
Species	Text	50	Yes	No	
Lifestage	Text	50	Yes	No	
PresenceJAN	Numeric	2	No	No	
PresenceFEB	Numeric	2	No	No	
PresenceMAR	Numeric	2	No	No	

PresenceAPR	Numeric	2	No	No	
PresenceMAY	Numeric	2	No	No	
PresenceJUN	Numeric	2	No	No	
PresenceJUL	Numeric	2	No	No	
PresenceAUG	Numeric	2	No	No	
PresenceSEP	Numeric	2	No	No	
PresenceOCT	Numeric	2	No	No	
PresenceNOV	Numeric	2	No	No	
PresenceDEC	Numeric	2	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
Recordnumber	Numeric	4	No	No	
presentInDrainage	Boolean	1	No	No	
segment	Text	50	Yes	No	

Table: FishPeriodicityPH
Description: Fish Periodicity/Presence data for each month, where 0 is no presence, 1 is present, and 2 is critical presence. Data in this table is specifically for the species in the Presumed Historic dataset.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	75	Yes	No	
Species	Text	50	Yes	No	
Lifestage	Text	50	Yes	No	
PresenceJAN	Numeric	2	No	No	
PresenceFEB	Numeric	2	No	No	
PresenceMAR	Numeric	2	No	No	
PresenceAPR	Numeric	2	No	No	
PresenceMAY	Numeric	2	No	No	
PresenceJUN	Numeric	2	No	No	
PresenceJUL	Numeric	2	No	No	
PresenceAUG	Numeric	2	No	No	
PresenceSEP	Numeric	2	No	No	
PresenceOCT	Numeric	2	No	No	
PresenceNOV	Numeric	2	No	No	
PresenceDEC	Numeric	2	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Recordnumber	Numeric	4	No	No	
presentInDrainage	Boolean	1	No	No	
segment	Text	50	Yes	No	

Table: FishUtilizationRestoration
Description: Text blocks of data for each drainage, describing fish utilization and any fish restoration priorities within that drainage.

Field	Type	Size	Key Field	Primary Key	Comments
UtilizationRestorationPriorities	Text	2048	No	No	
Drainage	Text	75	Yes	No	
Recordnumber	Numeric	4	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Glossary
Description: This is the glossary appearing at the end of the report; for each term, a definition is provided. The contents of this table are listed in alphabetic order in the glossary.

Field	Type	Size	Key Field	Primary Key	Comments
Term	Text	75	Yes	No	
Definition	Text	350	Yes	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
recordnumber	Numeric	4	Yes	Yes	

Table: InstreamFlow
Description: Contains legally established and proposed Instream Flow requirements for the given stream segment, which lies in the given drainage.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	75	Yes	No	
StreamSegment	Text	75	Yes	No	
LegallyEstablished	Boolean	1	No	No	
Proposed	Boolean	1	No	No	
DateLegal	Date/Time	8	No	No	
DateProposed	Date/Time	8	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

recordnumber	Numeric	4	Yes	Yes	
LegalJan	Numeric	8	No	No	
LegalFeb	Numeric	8	No	No	
LegalMar	Numeric	8	No	No	
LegalApr	Numeric	8	No	No	
LegalMay	Numeric	8	No	No	
LegalJune	Numeric	8	No	No	
LegalJuly	Numeric	8	No	No	
LegalAug	Numeric	8	No	No	
LegalSep	Numeric	8	No	No	
LegalOct	Numeric	8	No	No	
LegalNov	Numeric	8	No	No	
LegalDec	Numeric	8	No	No	
ProposedJan	Numeric	8	No	No	
ProposedFeb	Numeric	8	No	No	
ProposedMar	Numeric	8	No	No	
ProposedApr	Numeric	8	No	No	
ProposedMay	Numeric	8	No	No	
ProposedJun	Numeric	8	No	No	
ProposedJul	Numeric	8	No	No	
ProposedAug	Numeric	8	No	No	
ProposedSep	Numeric	8	No	No	
ProposedOct	Numeric	8	No	No	
ProposedNov	Numeric	8	No	No	
ProposedDec	Numeric	8	No	No	

Table: LayerData-DataTags
Description: Contains the layer tags from the project file for those layers which are used to extract data; the DataLayer field contains a textual description such as “Water Quality Sampling Sites” that the plug-in is looking for. TAG is the project file tag of the corresponding layer.

Field	Type	Size	Key Field	Primary Key	Comments
DataLayer	Text	50	Yes	Yes	
TAG	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: LayerData-MapTags
Description: Contains the project file tags for the layers going into each map, as well as the order that the layer appears in inside that map.

Field	Type	Size	Key Field	Primary Key	Comments
MapName	Text	50	Yes	No	Name of the map being generated; Plug-in looks for this.
TAG	Text	70	Yes	No	Project file tag of the layer going into the map.
OrderInMap	Numeric	4	No	No	Order that the layer will appear in the map.
recordnumber	Numeric	4	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: MonthName
Description: Contains the number of each month and the textual name, for lookup from reports.

Field	Type	Size	Key Field	Primary Key	Comments
MonthNum	Numeric	4	No	No	
MonthName	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Overview
Description: Contains the “Report Overview” text that appears near the front of the report. This table contains only one row of data.

Field	Type	Size	Key Field	Primary Key	Comments
Overview	Text	2147483647	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: PredefinedAggregations
Description: Contains the name of the drainage and the shapeindex of that drainage in the Watershed Delineation shapefile, along with the name of the aggregation that the drainage belongs to.

Field	Type	Size	Key Field	Primary Key	Comments
recordnum	Numeric	4	Yes	Yes	

AggregationName	Text	50	Yes	No	[Note duplication of data violating normal form; not worth overhead to have a separate (“aggregations”) table.]
DrainageName	Text	50	Yes	No	
ShapeIndex	Numeric	8	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Recreation
Description: Contains information regarding recreational activities for each drainage; the location in question, the activities available at that location, whether the activities include contact and noncontact activities, and the drainage the location falls within.

Field	Type	Size	Key Field	Primary Key	Comments
inDrainage	Text	75	Yes	No	Used to link to Drainages table.
Location	Text	75	Yes	No	
Activities	Text	250	No	No	
Contact	Boolean	1	No	No	
Noncontact	Boolean	1	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: ReportFirstPage
Description: Contains the data to be placed on the first page of the report. This table has only one row of data, and thus needs no primary key or other keys.

Field	Type	Size	Key Field	Primary Key	Comments
FirstLine	Text	255	No	No	The two smaller lines above
SecondLine	Text	255	No	No	The main (bigger font) label.
MainTitleSubnote	Text	255	No	No	Note to display below “Watershed Characterization Report”
FootnoteLine1	Text	255	No	No	The four lines of the footnote.
FootnoteLine2	Text	255	No	No	These may be left blank if they
FootnoteLine3	Text	255	No	No	are not needed.
FootnoteLine4	Text	255	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: SectionCommentary
Description: Table to hold report section comments.

Field	Type	Size	Key Field	Primary Key	Comments
SectionLink	Number	4	No	No	
Commentary	Ntext	16	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: SectionVisibility
Description: This table controls the visibility of individual report sections. When the report is generated, all sections are hidden – only those sections listed in this table are made visible again. This allows customization of the report (to hide sections) by editing this table.

Field	Type	Size	Key Field	Primary Key	Comments
sectionLink	Numeric	4	No	No	Corresponds to the sectionLink number in the crystal reports; also to the sectionLink in DataSources. One sectionLink generally exists for each report “section”.
isDisplayed	Boolean	1	No	No	True/false whether section is to be displayed. If no corresponding record is found in this table, false is assumed.
recordnumber	Numeric		No	Yes	
inSubreport	Text	75	No	No	Name of the Crystal report that this section appears in.
sectionsOfSubreport	Text	10	No	No	Crystal section numbers of sections in report that need to be shown for this sectionLink (ie, this section). If all sections in the report file are to be displayed, place -1 here.
requiredHeadingSectionNo	Numeric	4	No	No	Section number of any required heading; e.g. if section 1.1 is displayed, then section 1.1 requires the section that has the 1.0 label. (May be null if desired)
requiredHeadingSubreport	Text	75	No	No	Crystal Report name of the report where the required heading section occurs.

					(May be null if desired)
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: ShellFishbyDrainage
Description: Contains a text block of data regarding shellfish harvesting activities for a given drainage.

Field	Type	Size	Key Field	Primary Key	Comments
DrainName	Text	255	Yes	No	
ShellFish	Text	255	No	No	
recordnumber	Numeric	4	Yes	Yes	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: StreamClosures
Description: Contains closure status and closure period information for a given source and tributary. Includes a "LocatedInDrainage" field that's used to select this data from the report.

Field	Type	Size	Key Field	Primary Key	Comments
SourceName	Text	255	No	No	
TributaryTo	Text	255	No	No	
FormerAdministrativeStatus	Text	255	No	No	
StatusUnderRegulation	Text	255	No	No	
PeriodofClosure	Text	255	No	No	
recordnumber	Numeric	4	Yes	Yes	
LocatedInDrainage	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_FishDistComplete
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains the Fish Distribution data from the FishDist table, but formatted in such a way that the report may easily read it.

Field	Type	Size	Key Field	Primary Key	Comments
DrainageName	Text	50	No	No	
Species	Text	50	No	No	
Known	Text	6	No	No	

Presumed	Text	6	No	No	
Potential	Text	6	No	No	
Artificial	Text	6	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_Report303d
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds data regarding listed water bodies, copied from the 303d table. This table contains only the 303(d) data that falls within the boundaries of the drainages being reported on, where the 303d table contains *all* 303(d) data.

Field	Type	Size	Key Field	Primary Key	Comments
wType	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
wcrsNumber	Text	50	No	No	
Parameter	Text	50	No	No	
Medium	Text	75	No	No	
303d	Text	50	No	No	
FirstListedYear	Text	50	No	No	
Basis	Text	1024	No	No	
Remarks	Text	1024	No	No	
ActionNeeded	Text	50	No	No	
wbid	Text	50	No	No	
wgclNumber	Text	50	No	No	

Table: tmp_ReportClimate
Description: Table to hold summary information on climate data to report monthly average values of climate data used in the scenario simulation.

Field	Type	Size	Key Field	Primary Key	Comments
JANavgval	Float	8	No	No	January Average Value
Febavgval	Float	8	No	No	February Average Value
MARavgval	Float	8	No	No	March Average Value
APRavgval	Float	8	No	No	April Average Value
MAYavgval	Float	8	No	No	May Average Value
JUNavgval	Float	8	No	No	June Average Value
JULavgval	Float	8	No	No	July Average Value

SEPavgval	Float	8	No	No	September Average Value
OCTavgval	Float	8	No	No	October Average Value
NOVavgval	Float	8	No	No	November Average Value
DECavgval	Float	8	No	No	December Average Value
station	Text	75	Yes	No	Station ID
parameter	Number	4	No	No	Climate Variable
numobservations	Number	8	No	No	Number of Observations
period	Text	75	No	No	Time period
AUGavgval	Float	8	No	No	August

Table: tmp_ReportColiform
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds the number of samples and average sample value for fecal coliforms at each sampling site, taken from the WaterQuality database.

Field	Type	Size	Key Field	Primary Key	Comments
numColSamp	Numeric	8	No	No	
AveColSamp	Numeric	8	No	No	
stationname	Text	75	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportDataSources
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table has one data row only; it's got a field for each possible sectionLink value. When the report reads the DataSource information, it needs to be from a table having only one record for the links to work properly; so, this table is the data from DataSources, reformatted. The numeric field names correspond to the DataSource values for the sectionLink value of the field name. The fields that are numeric but beginning with A are the section heading labels for the sectionLink of the field's name. Numbers 100+ are the major section headings. No keys or indexes are necessary, as there is only one row.

Field	Type	Size	Key Field	Primary Key	Comments
1	Text	255	No	No	
11	Text	255	No	No	
12	Text	255	No	No	
13	Text	255	No	No	
14	Text	255	No	No	

15	Text	255	No	No	
16	Text	255	No	No	
17	Text	255	No	No	
18	Text	255	No	No	
19	Text	255	No	No	
20	Text	255	No	No	
2	Text	255	No	No	
3	Text	255	No	No	
4	Text	255	No	No	
5	Text	255	No	No	
6	Text	255	No	No	
7	Text	255	No	No	
8	Text	255	No	No	
9	Text	255	No	No	
10	Text	255	No	No	
21	Text	255	No	No	
22	Text	255	No	No	
23	Text	255	No	No	
24	Text	255	No	No	
25	Text	255	No	No	
26	Text	255	No	No	
27	Text	255	No	No	
28	Text	255	No	No	
29	Text	255	No	No	
30	Text	255	No	No	
31	Text	255	No	No	
32	Text	255	No	No	
33	Text	255	No	No	
34	Text	255	No	No	
35	Text	255	No	No	
36	Text	255	No	No	
37	Text	255	No	No	
38	Text	255	No	No	
39	Text	255	No	No	
40	Text	255	No	No	

41	Text	255	No	No	
42	Text	255	No	No	
43	Text	255	No	No	
44	Text	255	No	No	
45	Text	255	No	No	
46	Text	255	No	No	
47	Text	255	No	No	
48	Text	255	No	No	
49	Text	255	No	No	
50	Text	255	No	No	
51	Text	255	No	No	
52	Text	255	No	No	
53	Text	255	No	No	
A1	Text	255	No	No	
A2	Text	255	No	No	
A3	Text	255	No	No	
A4	Text	255	No	No	
A5	Text	255	No	No	
A6	Text	255	No	No	
A7	Text	255	No	No	
A8	Text	255	No	No	
A9	Text	255	No	No	
A10	Text	255	No	No	
A11	Text	255	No	No	
A12	Text	255	No	No	
A13	Text	255	No	No	
A14	Text	255	No	No	
A15	Text	255	No	No	
A16	Text	255	No	No	
A17	Text	255	No	No	
A18	Text	255	No	No	
A19	Text	255	No	No	
A20	Text	255	No	No	
A21	Text	255	No	No	
A22	Text	255	No	No	

A23	Text	255	No	No	
A24	Text	255	No	No	
A25	Text	255	No	No	
A26	Text	255	No	No	
A27	Text	255	No	No	
A28	Text	255	No	No	
A29	Text	255	No	No	
A30	Text	255	No	No	
A31	Text	255	No	No	
A32	Text	255	No	No	
A33	Text	255	No	No	
A34	Text	255	No	No	
A35	Text	255	No	No	
A36	Text	255	No	No	
A37	Text	255	No	No	
A38	Text	255	No	No	
A39	Text	255	No	No	
A40	Text	255	No	No	
A41	Text	255	No	No	
A42	Text	255	No	No	
A43	Text	255	No	No	
A44	Text	255	No	No	
A45	Text	255	No	No	
A46	Text	255	No	No	
A47	Text	255	No	No	
A48	Text	255	No	No	
A49	Text	255	No	No	
A50	Text	255	No	No	
A51	Text	255	No	No	
A52	Text	255	No	No	
A53	Text	255	No	No	
100	Text	255	No	No	
101	Text	255	No	No	
102	Text	255	No	No	
103	Text	255	No	No	

104	Text	255	No	No	
105	Text	255	No	No	
106	Text	255	No	No	
107	Text	255	No	No	
108	Text	255	No	No	
109	Text	255	No	No	
110	Text	255	No	No	
111	Text	255	No	No	
112	Text	255	No	No	
113	Text	255	No	No	
114	Text	255	No	No	
115	Text	255	No	No	
116	Text	255	No	No	
117	Text	255	No	No	
118	Text	255	No	No	
119	Text	255	No	No	
120	Text	255	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportFishPeriodicity
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table stores an image with the Lifestage Periodicity information, for each species, stream segment, and drainage. The image is generated by the FishPeriodicityPlotter control.

Field	Type	Size	Key Field	Primary Key	Comments
drainage	Text	75	No	No	
segment	Text	75	No	No	
species	Text	75	No	No	
PeriodicityImage	Binary	2147483647	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
recordnumber	Numeric	4	No	No	

Table: tmp_ReportFlowData

Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds the stream flow data summarized from the FlowData database, but only the data that falls within the bounds of the drainages being reported on. The Streamflow Sampling Sites shapefile is used to determine which station numbers are in the bounds of the drainages.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	50	No	No	
Agency_Code	Text	50	No	No	
Date	Date/Time	4	No	No	
Flow	Numeric	8	No	No	
Comments	Text	50	No	No	
RecordNum	Numeric	4	Yes	Yes	
Station Name	Text	255	No	No	
Location	Text	255	No	No	
Latitude	Numeric	8	No	No	
Longitude	Numeric	8	No	No	
Elevation	Numeric	4	No	No	
HUC	Numeric	4	No	No	
Base Flow	Numeric	8	No	No	
Drainage Area	Numeric	8	No	No	
Period	Text	50	No	No	
Years of Reference	Text	50	No	No	
DrainName	Text	50	No	No	
StreamName	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportFlowStationInfo

Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds information regarding the flow data stations in the bounds of the reporting drainages, summarized from the FlowData database.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	50	No	No	
Streamname	Text	50	No	No	
Station	Text	50	No	No	

StationName	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportingDrainages
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table stores a list of the drainage names that are being reported on; used by the report to link to the tables which use Drainage or DrainName as a selector.

Field	Type	Size	Key Field	Primary Key	Comments
Drainage	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportLandCover
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains each of the Land Cover types (from National Land Cover Dataset Classification System), along with the percentage of the selected drainages they cover and the total area reporting on.

Field	Type	Size	Key Field	Primary Key	Comments
totalarea	Numeric	4	No	No	This field is repeating information, but it makes the report calculations much easier to perform keeping it in this un-normalized form.
Level	Text	50	No	No	
percent	Numeric	8	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportNutrients
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds the number of samples and average sample value for each nutrient at each sampling site, taken from the WaterQuality database.

Field	Type	Size	Key Field	Primary Key	Comments
stationname	Text	75	No	No	
nutrient	Text	75	No	No	
meanvalue	Numeric	8	No	No	
numsamples	Numeric	8	No	No	

LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportPassData
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table has only one row of data, similar to ReportFirstPage. Stores miscellaneous data, such as report generator version, which periodicity dataset is in use, what units are selected, et cetera. As the table has one row, no keys are necessary.

Field	Type	Size	Key Field	Primary Key	Comments
ReportGenVersion	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
PeriodicityDataset	Text	25	No	No	
FlowUnits	Text	25	No	No	
VolumeUnits	Text	25	No	No	
YearDisplay	Text	25	No	No	
DisplayCoastalData	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportSectionCommentary
Description: Gathers the section commentary data from the SecionCommentary table and formats them into the one-row setup required for the report, placing that row into tmp_ReportSectionCommentary.

Field	Type	Size	Key Field	Primary Key	Comments
LocallyModified	Boolean	1	No	No	Locally modified or created.
100	Text	1024	No	No	
101	Text	1024	No	No	
102	Text	1024	No	No	
103	Text	1024	No	No	
104	Text	1024	No	No	
105	Text	1024	No	No	
106	Text	1024	No	No	
107	Text	1024	No	No	
108	Text	1024	No	No	
109	Text	1024	No	No	
110	Text	1024	No	No	
111	Text	1024	No	No	
112	Text	1024	No	No	

113	Text	1024	No	No	
114	Text	1024	No	No	
115	Text	1024	No	No	
116	Text	1024	No	No	
117	Text	1024	No	No	
118	Text	1024	No	No	
119	Text	1024	No	No	
120	Text	1024	No	No	

Table: tmp_ReportSocioeconomicData
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. Stores the number of employed population for each socioeconomic category.

Field	Type	Size	Key Field	Primary Key	Comments
category	Text	250	No	No	
value	Numeric	4	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
recordnumber	Numeric	4	No	No	

Table: tmp_ReportSwapImages
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains the map images used throughout the report; reportSwappedFor indicates the subreport for which this image was stored. The addtnlData fields are used in the Stream Flow report, and tell the report what stream segment, drainage, etc. the flow graph was stored for.

Field	Type	Size	Key Field	Primary Key	Comments
picturedata	Binary	2147483647	No	No	
size	Numeric	4	No	No	
filename	Text	50	No	No	
reportSwappedFor	Text	50	No	No	
addtnlData	Text	50	No	No	
addtnldata2	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
addtnldata3	Text	50	No	No	
addtnldata4	Text	50	No	No	

Table: tmp_ReportSwapImages2
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains the map images used throughout the report; reportSwappedFor indicates the subreport for which this image was stored. This table is used rather than tmp_ReportSwapImages in the case that more than one image is needed in the same subreport.

Field	Type	Size	Key Field	Primary Key	Comments
picturedata	Binary	2147483647	No	No	
size	Numeric	4	No	No	
filename	Text	50	No	No	
reportSwappedFor	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
addtnlData	Text	50	No	No	
addtnlData2	Text	50	No	No	
addtnlData3	Text	50	No	No	
addtnlData4	Text	50	No	No	

Table: tmp_ReportSwapImages3
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains the map images used throughout the report; reportSwappedFor indicates the subreport for which this image was stored. This table is used rather than tmp_ReportSwapImages or tmp_ReportSwapImage2 in the case that more than one image is needed in the same subreport.

Field	Type	Size	Key Field	Primary Key	Comments
picturedata	Binary	2147483647	No	No	
size	Numeric	4	No	No	
filename	Text	50	No	No	
reportSwappedFor	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.
addtnlData	Text	50	No	No	
addtnlData2	Text	50	No	No	
addtnlData3	Text	50	No	No	
addtnlData4	Text	50	No	No	

Table: tmp_ReportSwapNASBLG
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains map-related images – scale bar, legend. The name comes from NorthArrowScaleBarLeGend, before North Arrow was a static image on the report. reportSwappedFor indicates the subreport the images have been saved for.

Field	Type	Size	Key Field	Primary Key	Comments
scalebar	Binary	2147483647	No	No	
filename	Text	50	No	No	
reportSwappedFor	Text	50	No	No	
scalebar_size	Numeric	16	No	No	
legendLeft	Binary	2147483647	No	No	
legendRight	Binary	2147483647	No	No	
legendLeftSize	Numeric	16	No	No	
legendRightSize	Numeric	16	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportSwapNASBLG2
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table contains map-related images such as scale bar and legends. reportSwappedFor indicates the subreport the images have been saved for. This is used rather than tmp_ReportSwapNASBLG in the case that more than one map appears in the same report.

Field	Type	Size	Key Field	Primary Key	Comments
scalebar	Binary	2147483647	No	No	
filename	Text	50	No	No	
reportSwappedFor	Text	50	No	No	
scalebar_size	Numeric	4	No	No	
legendLeft	Binary	2147483647	No	No	
legendRight	Binary	2147483647	No	No	
legendLeftSize	Numeric	16	No	No	
legendRightSize	Numeric	16	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportTableOfContents
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This holds the table of contents data after it's been generated. Group is the section name, pagenumber is the page where the first page of the section appears, and inOrd is an ordering indicator.

Field	Type	Size	Key Field	Primary Key	Comments
Group	Text	100	No	No	
PageNumber	Text	50	No	No	
inOrd	Numeric	4	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportTemperature
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds the number of samples and average sample value for temperature at each sampling site, taken from the WaterQuality database.

Field	Type	Size	Key Field	Primary Key	Comments
numTempSamp	Numeric	8	No	No	Number of Temperature Samples
AveTempSamp	Numeric	8	No	No	Average sample value for temperature
stationname	Text	75	No	No	Station Name
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportWaterRights
Description: Temporary database table used to store flow data for each drainage during a scenario simulation.

Field	Type	Size	Key Field	Primary Key	Comments
inDrainage	Text	75	No	No	
KEY_SOURCE	Text	75	Yes	No	
PRIORITY_D	Text	50	No	No	Priority Date
DOCUMENT_T	Text	50	No	No	
KEY_MAIN	Text	75	Yes	No	
SOURCE	Text	50	No	No	Source
PURPOSE_LI	Text	50	No	No	
LAST_NAME	Text	75	No	No	Last Name
FIRST_NAME	Text	75	No	No	First Name

BUSINESS_N	Text	75	No	No	Business Name
CFS	Float	8	No	No	Cubic feet per second
GPM	Float	8	No	No	Gallons per minute
ACRE_FEET	Float	8	No	No	Acre feet
ACRES_IRR	Float	8	No	No	Acres irrigated
RELATED_DO	Text	75	No	No	
Locally_Modified	Boolean	1	No	No	Locally modified or created.

Table: tmp_ReportWaterUse
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table holds the water use information for all of the reporting drainages, combined and summarized from the WaterUse-* tables.

Field	Type	Size	Key Field	Primary Key	Comments
Flow	Numeric	8	No	No	Flow
Precip	Numeric	8	No	No	Precipitation
IndUse	Numeric	8	No	No	Landuse
Evap	Numeric	8	No	No	Evapotranspiration
AgUse	Numeric	8	No	No	Agricultural Use
DomMun	Numeric	8	No	No	Domestic Municipal Use
month	Numeric	4	No	No	Month of year

Table: tmp_reportZoning
Description: All tmp_ tables are used for internal processing in the report, and have their contents deleted at the beginning of each report run. This table stores the area of each zone in the selected drainages, and the name of that zone. Total area of the drainages for which zoning data is known is also stored.

Field	Type	Size	Key Field	Primary Key	Comments
area	Numeric	8	No	No	
zoneName	Text	50	No	No	
totalarea	Numeric	8	No	No	This is repeating data, but storing it makes report calculations much easier despite its deviance from normal forms.

Table: WaterSupplySystems
Description: Stores the name, info, and contact info for each water supply system, along with the drainage where the supply system lives.

Field	Type	Size	Key Field	Primary Key	Comments
SystemName	Text	50	No	No	Water Supply System
PopulationServed	Numeric	8	No	No	Number of People served
NumberConnections	Numeric	4	No	No	Number of connections
ContactName	Text	20	No	No	Water Supply Contact
ContactAddr	Text	30	No	No	Water Supply Address
ContactCityStateZip	Text	30	No	No	City State Zip
ContactPhone	Text	13	No	No	Phone number
Drainage	Text	75	No	No	Drainage Name
recordnumber	Numeric	4	No	No	Record number
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: WaterUse-Agricultural
Description: Stores the water use for agricultural purposes, by month and by drainage.

Field	Type	Size	Key Field	Primary Key	Comments
AgUse	Numeric	8	No	No	Agricultural Use
month	Numeric	4	No	No	Month of year
recordnumber	Numeric	4	No	No	Record number
LocallyModified	Boolean	1	No	No	Locally modified or created.
Drainage	Text	75	No	No	Drainage Name

Table: WaterUse-DomesticMunicipal
Description: Stores the water use for domestic and municipal purposes, by month and by drainage.

Field	Type	Size	Key Field	Primary Key	Comments
DomMun	Numeric	8	No	No	Domestic Municapal Use
month	Numeric	4	No	No	Month of year
recordnumber	Numeric	4	No	No	Record Number
LocallyModified	Boolean	1	No	No	Locally modified or created.
Drainage	Text	75	No	No	Drainage Name

Table: WaterUse-Evapotranspiration
Description: Stores the water loss to evapotranspiration, by month and by drainage.

Field	Type	Size	Key Field	Primary Key	Comments
Evap	Numeric	8	No	No	Evapotranspiration
month	Numeric	4	No	No	Month of year
recordnumber	Numeric	4	No	No	Record Number
LocallyModified	Boolean	1	No	No	Locally modified or created.
Drainage	Text	75	No	No	Drainage Name

Table: WaterUse-Industrial
Description: Stores the water use for industrial purposes, by month and by drainage.

Field	Type	Size	Key Field	Primary Key	Comments
IndUse	Numeric	8	No	No	Industrial Use
month	Numeric	4	No	No	Month of year
recordnumber	Numeric	4	No	No	Record Number
LocallyModified	Boolean	1	No	No	Locally modified or created.
Drainage	Text	75	No	No	Drainage Name

Table: WaterUse-Precipitation
Description: Stores the water gain from precipitation, by month and by drainage.

Field	Type	Size	Key Field	Primary Key	Comments
Precip	Numeric	8	No	No	Precipitation
month	Numeric	4	No	No	Month of year
recordnumber	Numeric	4	No	No	Record Number
LocallyModified	Boolean	1	No	No	Locally modified or created.
Drainage	Text	75	No	No	Drainage Name

Table: WellsbyDrainage
Description: Contains a count of wells and a well-related comment for each drainage.

Field	Type	Size	Key Field	Primary Key	Comments
DrainName	Text	255	No	No	Drainage Name
WellsComment	Text	100	No	No	Well-related comment
NumWells	Numeric	4	No	No	Number of wells in drainage
recordnumber	Numeric	4	No	No	Record Number
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: ZoningAbbreviations
Description: Stores the zoning abbreviations used in the Zoning shapefiles; abbreviation is the shorthand notation, and fullname is the full name to be printed in the report.

Field	Type	Size	Key Field	Primary Key	Comments
Abbreviation	Text	50	No	No	Zoning Abbreviation
FullName	Text	50	No	No	Zoning Description
RecordNumber	Numeric	4	Yes	Yes	Record Number
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: zwatbalcfs
Description: Legacy summary water balance data (cfs). Not used by the existing watershed characterization program.

Field	Type	Size	Key Field	Primary Key	Comments
AREA	Float	8	No	No	Area
PERIMTER	Float	8	No	No	Perimeter
BSNSWRIA1_	Float	8	Yes	No	WRIA unique ID
POLY_	Float	8	No	No	GIS ID
SUBCLASS	Text	255	No	No	
SUBCLASS_	Float	8	No	No	
RINGS_OK	Float	8	No	No	GIS topology check
RINGS_NOK	Float	8	No	No	GIS topology check
DRAIN3_	Float	8	No	No	
DRAIN3_ID	Float	8	No	No	
NAME	Text	255	No	No	
DRAINAGE	Float	8	No	No	
WSHED	Float	8	No	No	
SUBWSHED	Float	8	No	No	
REGION	Text	255	No	No	
SUBBASIN	Text	255	No	No	
GROUP_NAME	Text	255	No	No	
ACRES	Float	8	No	No	Area expressed as Acres
DRAINS_TO	Text	255	No	No	
DRAIN_TYPE	Text	255	No	No	
JAN_NCU	Float	8	No	No	
FEB_NCU	Float	8	No	No	

MAR_NCU	Float	8	No	No	
ARP_NCU	Float	8	No	No	
MAY_NCU	Float	8	No	No	
JUN_NCU	Float	8	No	No	
JUL_NCU	Float	8	No	No	
AUG_NCU	Float	8	No	No	
SEP_NCU	Float	8	No	No	
OCT_NCU	Float	8	No	No	
NOV_NCU	Float	8	No	No	
DEC_NCU	Float	8	No	No	
TOT_NCU	Float	8	No	No	
ET_JAN	Float	8	No	No	Evapotranspiration January
ET_FEB	Float	8	No	No	Evapotranspiration February
ET_MAR	Float	8	No	No	Evapotranspiration March
ET_APR	Float	8	No	No	Evapotranspiration April
ET_MAY	Float	8	No	No	Evapotranspiration May
ET_JUNE	Float	8	No	No	Evapotranspiration June
ET_JULY	Float	8	No	No	Evapotranspiration July
ET_AUG	Float	8	No	No	Evapotranspiration August
ET_SEPT	Float	8	No	No	Evapotranspiration September
ET_OCT	Float	8	No	No	Evapotranspiration October
ET_NOV	Float	8	No	No	Evapotranspiration November
ET_DEC	Float	8	No	No	Evapotranspiration December
ET_ANN	Float	8	No	No	Annual Evapotranspiration
JANQN	Float	8	No	No	January
FEBQN	Float	8	No	No	February
MARQN	Float	8	No	No	March
APRQN	Float	8	No	No	April
MAYQN	Float	8	No	No	May
JUNQN	Float	8	No	No	June
JULQN	Float	8	No	No	July
AUGQN	Float	8	No	No	August
SEPQN	Float	8	No	No	September
OCTQN	Float	8	No	No	October
NOVQN	Float	8	No	No	November

DECQN	Float	8	No	No	December
ANNQN	Float	8	No	No	Annual
STDERR	Float	8	No	No	Standard Error
JANQNCFS	Float	8	No	No	January
FEBQNCFS	Float	8	No	No	February
MARQNCFS	Float	8	No	No	March
APRQNCFS	Float	8	No	No	April
MAYQNCFS	Float	8	No	No	May
JUNQNCFS	Float	8	No	No	June
JULQNCFS	Float	8	No	No	July
AUGQNCFS	Float	8	No	No	August
SEPQNCFS	Float	8	No	No	September
OCTQNCFS	Float	8	No	No	October
NOVQNCFS	Float	8	No	No	November
DECQNCFS	Float	8	No	No	December
ANNQNCFS	Float	8	No	No	Annual
JANQP	Float	8	No	No	January
FEBQP	Float	8	No	No	February
MARQP	Float	8	No	No	March
APRQP	Float	8	No	No	April
MAYQP	Float	8	No	No	May
JUNQP	Float	8	No	No	June
JULQP	Float	8	No	No	July
AUGQP	Float	8	No	No	August
SEPQP	Float	8	No	No	September
OCTQP	Float	8	No	No	October
NOVQP	Float	8	No	No	November
DECQP	Float	8	No	No	December
ANNQP	Float	8	No	No	Annual
JANQPCFS	Float	8	No	No	January
FEBQPCFS	Float	8	No	No	February
MARQPCFS	Float	8	No	No	March
APRQPCFS	Float	8	No	No	April
MAYQPCFS	Float	8	No	No	May
JUNQPCFS	Float	8	No	No	June

JULQPCFS	Float	8	No	No	July
AUGQPCFS	Float	8	No	No	August
SEPQPCFS	Float	8	No	No	September
OCTQPCFS	Float	8	No	No	October
NOVQPCFS	Float	8	No	No	November
DECQPCFS	Float	8	No	No	December
ANNQPCFS	Float	8	No	No	Annual
PCP_JAN	Float	8	No	No	January
PCP_FEB	Float	8	No	No	February
PCP_MAR	Float	8	No	No	March
PCP_APR	Float	8	No	No	April
PCP_MAY	Float	8	No	No	May
PCP_JUN	Float	8	No	No	June
PCP_JUL	Float	8	No	No	July
PCP_AUG	Float	8	No	No	August
PCP_SEP	Float	8	No	No	September
PCP_OCT	Float	8	No	No	October
PCP_NOV	Float	8	No	No	November
PCP_DEC	Float	8	No	No	December
PCP_ANN	Float	8	No	No	Annual
JANWBERR	Float	8	No	No	January Water Balance Error
FEBWBERR	Float	8	No	No	February Water Balance Error
MARWBERR	Float	8	No	No	March Water Balance Error
APRWBERR	Float	8	No	No	April Water Balance Error
MAYWBERR	Float	8	No	No	May Water Balance Error
JUNWBERR	Float	8	No	No	June Water Balance Error
JULWBERR	Float	8	No	No	July Water Balance Error
AUGWBERR	Float	8	No	No	August Water Balance Error
SEPWBERR	Float	8	No	No	September Water Balance Error
OCTWBERR	Float	8	No	No	October Water Balance Error
NOVWBERR	Float	8	No	No	November Water Balance Error
DECWBERR	Float	8	No	No	December Water Balance Error
WBERR	Float	8	No	No	Water Balance Error
WBERRPERC	Float	8	No	No	Water Balance Error (pct)
SWTOT	Float	8	No	No	Surface Water Total

GWTOT	Float	8	No	No	Groundwater Total
SWRES	Float	8	No	No	Surface Water Residential
GWRES	Float	8	No	No	Ground Water Residential
SWCI	Float	8	No	No	Surface Water Commercial Industrial
GWCI	Float	8	No	No	Groundwater Commercial Industrial
SWAG	Float	8	No	No	Surface Water Agriculture
GWAG	Float	8	No	No	Groundwater Agriculture
SWPWS	Float	8	No	No	
GWPWS	Float	8	No	No	
JANQDEFCFS	Float	8	No	No	January
FEBQDEFCFS	Float	8	No	No	February
MARQDEFCFS	Float	8	No	No	March
APRQDEFCFS	Float	8	No	No	April
MAYQDEFCFS	Float	8	No	No	May
JUNQDEFCFS	Float	8	No	No	June
JULQDEFCFS	Float	8	No	No	July
AUGQDCFS	Float	8	No	No	August
SEPQDEFCFS	Float	8	No	No	September
OCTQDEFCFS	Float	8	No	No	October
NOVQDEFCFS	Float	8	No	No	November
DECQDEFCFS	Float	8	No	No	December
ANNQDEFCFS	Float	8	No	No	Annual
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Parameter_Code Descriptions
Description: Describes type of sample, e.g. temperature, fecal coliform, etc.

Field	Type	Size	Key Field	Primary Key	Comments
Parameter_Code	Numeric		No	No	
Parameter_Name	Text	255	No	No	
Class	Text	50	No	No	
Common_Name	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: QAQC_Code Descriptions
Description: Quality Assurance/Quality Check Methods used to collect data, by QAQC code. [Not used by WRIA Report Generator.]

Field	Type	Size	Key Field	Primary Key	Comments
QAQC_Code	Text	50	No	No	
Description	Text	150	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Source_Database Descriptions
Description: Full data source descriptions for each source database. [Not used by WRIA Report Generator.]

Field	Type	Size	Key Field	Primary Key	Comments
Source_Database	Text	50	No	No	
Description	Text	255	No	No	
Source_Contact	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: Stations
Description: Station information, e.g. location, lat, long, subbasin, etc. for each station number.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	50	Yes	Yes	
Shapefile	Numeric		No	No	
USU_Subbasin	Text	50	No	No	
Station_Name	Text	100	No	No	
Location	Text	200	No	No	
State	Text	50	No	No	
County	Text	50	No	No	
Latitude	Numeric		No	No	
Longitude	Numeric		No	No	
UTMX	Numeric		No	No	
UTMY	Numeric		No	No	
River_Mile	Numeric		No	No	
Station_Type	Text	75	No	No	
NHDID	Text	50	No	No	
Wb_Name	Text	50	No	No	

WRIA1_Shed	Text	50	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

Table: WQData
Description: Actual Water Quality sampling data. Contains values sampled, with Parameter_Code as most useful field to look up what the value sampled is. Comments, QAQC, Data, Source all look up to appropriate tables.

Field	Type	Size	Key Field	Primary Key	Comments
Station	Text	255	Yes	No	
Agency_Code	Text	255	No	No	
Date	Date/Time		No	No	
Depth	Numeric		No	No	
Parameter_Code	Numeric		Yes	No	
Value	Numeric		No	No	
Comment_Code	Text	255	No	No	
QAQC_Code	Text	255	No	No	
Data_Code	Text	255	No	No	
Source_Database	Text	255	No	No	
LocallyModified	Boolean	1	No	No	Locally modified or created.

2. GIS Data Needs

MapWindow must be running with the WRIA-1 DSS project loaded. Inside this project, layers must have tags associated with them (through the project file). The names of the tags that need to be included in the project are defined in the database tables LayerData-DataTags and LayerData-MapTags.

The layers whose tags are listed in LayerData-MapTags are used only for producing maps, and do not have data extracted from them. These layers may be added and removed freely, as long as the LayerData-MapTags table is kept in sync with what's in the project file.

The layers whose tags are listed in LayerData-DataTags are those layers from which data will be extracted. The report needs to be able to find a layer for each of these data items; there must be a layer for each of these. The tag which corresponds to each of these is defined in the LayerData-DataTags table. Please see this table for a definitive list; a partial list is given below.

Required data layers and fields used from each:

1. Watershed (Shapefile)
    a. DRAIN_NAME – Name of drainage.
2. Zoning Data (Shapefile)
    a. WCPLAN – Zone Type, e.g. City, Rural, etc.
3. Streamflow Gages (Shapefile)
    a. SITEID – Sampling Site ID
    b. STR_NAME – Stream Name
4. Land Cover Data (Grid - Grid Value Used)
5. Main Roads (Shapefile - No Fields Used, but must be present)
6. Water Quality Stations
    a. STATION – Station ID Number
    b. STATION_NA – Station Name
7. Cities (Shapefile – No Fields Used, but must be present)
8. Railroads (Shapefile – No Fields Used, but must be present)
9. 303(d) Water Bodies (Shapefile)
    a. WTRBDY_NR – Waterbody Number
10. 303(d) Listed Streams (Shapefile)
    a. WTRCRS_NR – Watercourse Number
11. 303(d) Listed Grids (Shapefile)
    a. WGRD_CL_NR – Watergrid Cell Number
12. Counties (No Fields Used, but must be present)

3. Dependencies

The Watershed Characterization plug-in requires the following software components and modules to be installed:

Software:

MapWindow 3.1
Visual Studio .NET 2003 Complete Install
--Be sure to select Crystal Reports.Net for installation--
--Crystal Reports 9.0 may be used instead.--
GigaSoft ProEssentials 3.0 Charting Components
Microsoft SQL Server 2000 (optional)
--This will provide administration tools which make management of the databases much easier.--
InstallShield Express 3.5 with Service Pack 4

Components:

Microsoft Common Controls 6.0
Microsoft Common Controls-2 6.0
Microsoft Common Controls-3 6.0
Microsoft FlexGrid Control 6.0
Fish Periodicity Lifestage Plotter Control (USU)
Microsoft SQL-DMO Data Object

4. Setup

Setup

The LaunchPad utility, also referred to as the DBMS utility, may be used to prepare a computer to run the Watershed Characterization report. This tool is described in the DBMS Technical Documentation.

The report generator also requires that the WRIA-1 DSS project file is loaded in MapWindow. Details on this project file are given under section 3 above.

To launch the report generator, click the Graph icon on the toolbar. If the database is accessible and configured properly, the Launch Report screen will appear where you may specify report options and begin generation.

If the database could not be found or was not configured, a window will appear asking for the location of the database. This could be a Microsoft SQL Server, MSDE, or access databases. The database needs are as follows:

Database Description	SQL Server DB Name	Access Name (databases may share physical files, e.g. FlowDataWaterQuality.mdb)
Report Generator database, as described above	WRIAReportGenerator	<may be any .mdb file>
Stream Flow database, as described above	FlowData	<may be any .mdb file>
Water Quality database, as described above	WaterQuality	<may be any .mdb file>

Customizations

Customizations may be made to the report itself very easily; most text sections may be edited via the DBMS utility. Those tables which are not exposed through the DBMS may be edited via Microsoft SQL Server Enterprise Manager (a tool included with SQL Server 2000), or via any ODBC-compatible data access utility.

Adding a report section can be done in Visual Studio by following these instructions:

a. Create a new blank report. Save it with the desired filename. This filename will be used as the “reportSwappedFor” string below.
b. If the section has a map, place the “reportSwappedFor” string in the NoDataValidation routine inside DataGeneration.vb to ensure that a map image is always placed in the table at runtime.
c. Ensure that the sectionLink number is unique. This number is also used for the section headings. Place a line for this unique number in the DataSources table

with the section heading and name. The naming convention is that any number less than 100 is a subsection heading, and anything 100 or over is a main heading.

d. Place the sectionLink number in the SectionVisibility table along with the report section numbers that this report portion will use. Failure to do this will result in the section never appearing.
e. Place any data items in the report. DO NOT use visual linking; any selection formulas must be put by hand in the selection formula editor or selection formula assistant. If the report has a map, select from the data tables based on reportSwappedFor.
f. Open rpt_FullReport.rpt, and insert a new section where desired. Insert the new report here.
g. Add code to the DataGeneration.vb and/or ReportMain.vb source files to create a new map if necessary, and to generate or prepare any data needed.

Similary, removing a report section permanently can be done in Visual Studio by doing the reverse of the instructions above:

a. Make note of the sectionLink number(s) in the report. Delete the report.
b. Remove this subreport from rpt_FullReport.rpt.
c. Delete any code from DataGeneration.vb and/or ReportMain.vb that was used to prepare maps and data for this report.
d. Remove the records from SectionVisibility and DataSources where the sectionLink is equal to the number noted in step 1.

5. Building

Compiling the Watershed Characterization Report Generator is a fairly straightforward task. After ensuring that all of the required components discussed in item 3 above are present, load the project into Visual Studio and click the Build icon, or select Build from the menu. There are two projects which are a part of this solution. First, the Watershed Characterization MapWindow Plugin is the component loaded into MapWindow which generates all needed data. At the end of the report generation sequence, the plugin will then automatically start ReportViewer.exe, which is the second project in the solution. The report viewer will continue the process, loading crystal reports and preparing the final report. The Crystal Reports portion of the report must be done in a separate executable, because Crystal Reports will not load properly if it's loaded in the same process image as MapWindow, for unknown reasons. The solution needs to include the following files:

mwWatershedChar project:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.

Camera.ico	Camera icon representing snapshots.
Datageneration.vb	Contains data generation routines and routines to aggregate data from multiple locations. Also contains calls to generate Streamflow charts and fish periodicity charts.
dataGeneration-StreamFlowBoxPlots.vb	Contains generation routines for Streamflow box charts. Mostly copied from StreamFlow Analyst, but modified slightly to fit in this plug-in.
DataLayerManager.vb	Contains a class to keep track of data layers within maps by tag or by layer handle.
Datapuzzle.avi	File puzzle animation used on the progress meter screen.
Drip16.bmp	Drip icon for stream sampling locations on maps.
Emptylegend.bmp	An empty (white) bitmap, properly sized to be inserted in place of a legend should a spacer be needed.
Flask16.bmp	Flask icon used for water quality sampling sites on maps.
frmConfiguration.vb	Screen with database connection information, used to specify the location of the SQL or Access databases being used.
frmDataGenProgress.vb	Displays a progress meter and animation, along with a brief text description of what's happening.
frmLauncher.vb	Displays the list of aggregations and allows the user to specify options for the report. The report is launched from here.
frmMapGen.vb	Contains a MapWinGIS Map control. The maps for the reports are generated on this map using this class.
frmPeriodicityPlotter.vb	Contains a Fish Periodicity Plotter control; this is used by the dataGeneration class to produce images.
frmReportGraphGenerator.vb	Contains Gigasoft graphing components.
frmSectionVisibility.vb	Allows the user to change which report sections are going to be included in the report.
Globals.vb	This module holds some commonly used functions such as ComputeSimpleArea.
Graph.ico	This is the graph icon used on the toolbar.
ImageConverter.vb	Performs conversions between old-style IPictureDispatch objects and newer system.drawing.bitmap objects.
LabelClass.vb	This class is used to place labels on the maps as they are generated. This was originally copied form MapWindow, but has been extensively modified.
Main.vb	Contains the implementation of the MapWindow Interface. Also contains shared data elements and shared functions.
Nodata-*.bmp	These files contain empty images, with the text "No data available". These are inserted into the report when data is unavailable, so that the report sees data and can link everything together.
North_arrow2.gif	This is the north arrow placed on the corner of maps.

ReportMain.vb	This contains all of the preparation the reports themselves. Maps are generated in this module as well.
ScaleBarUtils.vb	Copied from MapWindow, this module contains conversion routines and factors (ie, inch to millimeter, etc). This also contains routines to generate accurate scale bars.

ReportViewer.exe Project:

frmReportView.vb	Contains a CrystalReportViewer object, used to examine the report if it's generated to screen rather than to a file.
rpt_FullReport.rpt	This contains page numbering formulas, report header information, and ties together all of the other reports (by pulling them in as subreports). Sections are hidden and made visible automatically from this report.
rpt_Shellfish-NoData.rpt	This report is shown in place of rpt_Shellfish-1 and rpt_Shellfish-2 when no shoreline is present in the selected drainages.
rpt_TOCGenerator.rpt	This report summarizes all Table of Contents related data gathered during the first report execution. This data is written in a manner that may be easily parsed, so it may be read by the generator before the final execution.
Rpt_*	Any rpt_*.rpt file not described above contains data used in the report. Section contents are described in the DataSources and SectionVisibility database tables.

The Watershed Characterization installer has been created with Microsoft Visual Studio, and it creates a "merge module", which is a single-file module which can then be inserted into other installations easily. This is the "mwWatershedChar" subdirectory in the Watershed Characterization installer directory.

To update the components in this merge module, copy the newly built mwWatershedChar.dll and related files into the Source directory in the Watershed Characterization installer directory. Open the mwWatershedChar Visual Studio project and rebuild the installer. Next, copy the merge module from the Release folder into "c:\Program Files\InstallShield\Express\Objects".

Another installation, this one created with InstallShield Express, is also located in the Watershed Characterization installer directory. This is a "wrapper" around the merge module described above; it installs no new files, only that merge module. To rebuild this, open the WatershedChar.ise file and click the "Build" button or select "Build" from the menu. When the process has completed, the new installer will be in the WatershedChar subdirectory.

This installer may then be burned onto a CD, copied to a ZIP disk, or otherwise distributed.

Technical Documentation: Well Log Data Viewer

Last Revision: 06/15/06

Table of Contents

1. *Table Design For Database Store Well Logs*
   - *Table Descriptions and Notes*
   - *Schema, Keys, Indexes and Structure*
   - *Table Relationship Diagrams*

2. *Data Needs*
   - *Type/Location of Data*
   - *Tags on Layers*

3. *Dependencies*
   - *Software*
   - *Modules and Components*
   - *Integrated Development Environment (IDE)*

4. *Setup*
   - *Setup*

5. *Code Compiling*

1. Table Design for Database Stored Well Logs

Following is a list of tables that must be included with the Well Log Viewer Database. Other tables and information may exist, but these tables must follow the described naming conventions, spelling and cases, and types for each table and its parameters. This database is only necessary if you are using/associating Database Stored Well Logs.

Table: Material_Codes
Description: Contains the ID, Name, and an Order for each of the valid Materials. It is important that it contains the complete set of valid Materials allowed for displaying the Well Logs because this table is linked to as a Lookup field from the Well_Material_Data table.

Field	Type	Size	Key Field	Primary Key	Comments
CodeMaterial	Numeric	Long Integer	Yes	Yes	A unique ID for each Well Material. It does not need to be in a strict numerical order. No Duplicates may be used. *For Example: 1, 2, 3, … , 20, etc.*
Material	Text	50	No	No	The name of the Material. This value is what is viewed in the Legend of the Viewer.
order	Text	50	No	No	An alphabetic value that is used to indicate an ordering for selecting or viewing the valid Materials. *For Example: a, b ,c,…, y, z ,etc.*

Table: Material_Colors
Description: Contains the associated material code, an RGB-value, and a description for each color. This table needs to have a 1:1 correlation with the Material_Codes table so that there is an assigned color for each valid material. These colors are used to display the Database Stored Well Logs in the Well Log Data Viewer.

Field	Type	Size	Key Field	Primary Key	Comments
CodeMaterial	Numeric	Long Integer	Yes (No Duplicates)	No	A unique ID for each Well Material. It corresponds with the CodeMaterial Field in the Material_Codes table. There needs to be one entry in this table for every entry in the Material_Codes table (a 1:1 correlation).

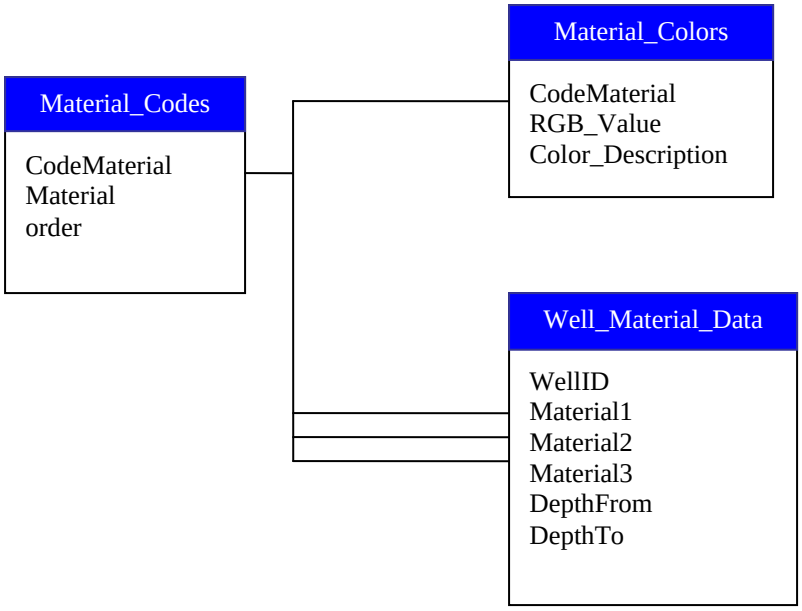
RGB_Value	Text	50	No	No	This is the RGB value (stored as R,G,B) for the color for the associated material. Each R,G, and B value ranges from 0 – 255. *For Example: RGB_Value = 0,0,0*
Color_Description	Text	50	No	No	This is the name of the color, or some other description of what color is represented by the value in the RGB_Value Field. *For Example: Color_Description = Black*

Table: Well_Material_Data
Description: Contains the WellID, three associated material codes, a depth from, and a depth to value for each soil section in the Well Log. A soil section is defined as a grouping of soil with the same materials. Each soil section in the sampled Well Log needs to have a unique entry in this Table. The second and/or third material code can be left empty if there is not a value attributed for them. The first material code must have a value for all entries. Each material code is picked from the list available that is populated by the values in the Material_Codes table.

Field	Type	Size	Key Field	Primary Key	Comments
WellID	Numeric	Long Integer	Yes (Duplicates OK)	No	A unique ID for each Well. This value is displayed for the user to select which Well Logs(s) to view. It also corresponds with the values for the Well_ID field in the associated point shapefile.
Material1	Number	Long Integer	Yes (Duplicates OK)	No	This is the main soil type, or the soil type with the highest concentration, that is found in the current soil section. This field is of type Number and corresponds to the CodeMaterial field values from the Material_Codes table. This is a lookup field. The user will see the material descriptions, not the actual material codes when selecting/inputting values into this field if it is set up correctly. See below for a definition of the Lookup Values.
Material2	Number	Long Integer	Yes (Duplicates OK)	No	This is the soil type with the second highest concentration of all of the materials in the current soil section.

					This field is just like Material1. It is of type Number and corresponds to the CodeMaterial field values from the Material_Codes table. If there is not a valid value for this field, it may be left empty. This is a lookup field. The user will see the material descriptions, not the actual material codes when selecting/inputting values into this field if it is set up correctly. See below for a definition of the Lookup Values.
Material3	Number	Long Integer	Yes (Duplicates OK)	No	This is the soil type with the third highest concentration of all of the materials in the current soil section. This field is just like Material1 and Material 2. It is of type Number, and corresponds to the CodeMaterial field values from the Material_Codes table. If there is not a valid value for this field, it may be left empty. This is a lookup field. The user will see the material descriptions, not the actual material codes when selecting/inputting values into this field, if it is set up correctly. See below for a definition of the Lookup Values.
MaterialFrom	Number	Double	No	No	This value is the starting depth of the current soil section. This value signifies the beginning depth. This value is stored in Feet (ft). *For Example: If you were entering the first soil section for your WellID, MaterialFrom = 0.*
MaterialTo	Number	Double	No	No	This value is the ending depth of the current soil section. This value signifies the ending depth. This value stored in Feet (ft). *For Example: If the current soil section started at 50ft and was 34ft deep, MaterialTo = 84.*

4881
4882


Material_Codes
CodeMaterial
Material
order
Material_Colors
CodeMaterial
RGB_Value
Color_Description
Well_Material_Data
WellID
Material1
Material2
Material3
DepthFrom
DepthTo

4883

4884

2. Data Needs

Scanned Well Log Images

There are two types of data needed for the Scanned Well Log Images: a point shapefile, and a Folder Location.

- Point Shapefile – this needs to be in the same projection as the other shapefiles in your project. It contains the point locations of where each of the nodes are located. This file will have a tag that will be associated with it by the plug-in: "WellLogViewer-Scan". There are two fields that need to be found in this shapefile: WELL_LOG_I, and LOG_IMG_NM.
- Folder Location – this is the location of all of the scanned Well Log images that will be viewed using the Well Log Data Viewer.

The layer corresponding with the point shapefile stated above is marked with the tag "WellLogViewer-Scan". If the user removes the layer associated with this tag without first closing the Well Log Data Viewer plug-in, then the settings for the Scanned Data will be reset, and if the Project is then saved, then the settings for the Scanned Data in the project file will also be over-written.

As stated above, the shapefile requires two fields to be found in the data table. Below are the exact field names, field types, and description of the data that gets entered into these fields.

Required data layer fields:

a. WELL_LOG_I – Integer – The Well ID for the Well at this location.
b. LOG_IMG_NM – String – The filename for the Scanned Well Log image that goes with this Well.

Database Data

There are three types of data needed for the Database Data: a point shapefile, a database type, and a database.

- Point Shapefile – this needs to be in the same projection as the other shapefiles in your project. It contains the point locations of where each of the Well Logs were sampled. This file will have a tag that will be associated with it by the plug-in: "WellLogViewer-DB". There is one field that needs to be found in this shapefile: WELL_ID.
- Database Type – this is the type of database that you are connecting to. IE: Access, SQL Server.
- Database – this is the database of Well Log data and needs to be organized as described in *Section 1. Table Design For Database Stored Well Logs.*

The layer corresponding with the point shapefile stated above is marked with the tag "WellLogViewer-DB". If the user removes the layer associated with this tag without first closing the Well Log Data Viewer plug-in, then the settings for the Database Data will be reset, and if the Project is then saved, then the settings for the Database Data in the project file will also be over-written.

As stated above, the shapefile requires one field to be found in the data table. Below are the exact field name, field type, description of the data that gets entered into this field, and how it is associated with the database.

Required data layer field:

a. WELL_ID – Integer – The Well ID for the Well at this location. It corresponds with the WellID field in the database. Each value that is found in this field (in the shapefile) also needs to be found in the database.

3. Dependencies

The Well Log Data Viewer plug-in requires the following software components and modules to be installed:

Software:

Gigasoft's ProEssentials Graphing Tools Version 3
MapWindow 3.1
Visual Studio .NET 2002

Modules and Components:

Graph Component from ProEssentials (pegoa control).
MapWindow Interfaces
MapWinGIS Map Control
Scientific Graph Component from ProEssentials (pesgoa control).

Integrated Development Environment (IDE)

Visual Studio .NET 2002

4. Setup

To begin using the Well Log Data Viewer, it must first be loaded into MapWindow. Once it is loaded, select one of your data layers from the Legend. If you only have one type of data associated, select the data layer that goes with this data. Now, click the Well icon, , on the MapWindow toolbar. If the data associations (as described in *Section 2: Data Needs*) have been properly set, then the Well Viewer will appear for the selected type of data, and will be fully functional.

If the data associations have not been set, then a Connection Form will appear allowing you to set the shapefile, database, and photo location folder. Then, after the data associations have been formed, select the appropriate data layer (as described above), and when the toolbar button is clicked, the viewer will show for the selected type of data, and can be used.

See the User's Manual for more information and details on associating the data with the Well Log Data Viewer.

5. Code Compiling

Compiling the Well Log Data Viewer is a fairly straightforward task. After ensuring that all of the required components discussed in *Section 3: Dependencies* are present, load the project into Visual Studio .Net 2002. This Plug-in was created using Visual Basic (VB).

The project needs to include the following files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsMainMWI.vb	Contains a class that implements the MapWindow plugin interface.
FolderUtils.vb	Contains a class to access global functions for Folder Functions, Database Functions, and Other Functions.
frmConfiguration.vb, frmConfiguration.resx	Displays the Connection form for the SQL database type. This form is only displayed when adding Database Data and connecting to an SQL database. It allows the user to set the Server Address, User information, and Database Name.
frmDBConnection.vb, frmDBConnection.resx	Displays the Connection form that allows the user to associate the data with the Well Log Data Viewer plug-in. There are two areas on this form, one for each type of data that can be associated: Scanned Data, and Database Data.
frmSelectLayer.vb, frmSelectLayer.resx	Displays the form to select the associated point shapefile with the Photo Viewer plug-in. It allows the user to either select a shapefile already loaded into Map Window, or to select one from disk.
frmWellViewer.vb, frmWellViewer.resx	Displays the Well Log Data Viewer form for the selected type of data. You select the type of data to view by selecting the appropriate data layer in the MapWindow Legend.
globalStructs.vb	This module holds the definition of data Structs used throughout the Well Log Data Viewer.
globalVariables.vb	This module holds the variables used throughout the forms, such as the form declaration variables, MapWindow variables, and others.
ImageConverter.vb	Contains a class that implements functions that allows you to convert images to and from an IPictureDisp object. This is needed so that the VBCompatibility.dll does not need to be referenced. This class allows the Well Logs – Database image, , or the Well Logs – Scanned image, , to be associated with the points on the associated shapefile(s).

PluginInfo.vb	Contains a class that has functions used to open and save Project File settings for the Well Log Data Viewer plugin.
Well.ico	Well Log icon that is used in the MapWindow Toolbar and on all of the forms. *NOTE: this file needs to be an embedded resource*
WellLog2.bmp	Bitmap version of the WellLog2.ico. It is used as a custom image for the point shapefile associated with the Database Data. *NOTE: this file needs to be an embedded resource*
WellLog2.ico	It is used as the MapWindow Legend picture when using a custom image for the point shapefile associated with the Database Data. *NOTE: this file needs to be an embedded resource*
WellReport2.bmp	Bitmap version of the WellReport2.ico. It is used as a custom image for the point shapefile associated with the Scanned Data. *NOTE: this file needs to be an embedded resource*
WellReport2.ico	It is used as the MapWindow Legend picture when using a custom image for the point shapefile associated with the Scanned Data. *NOTE: this file needs to be an embedded resource*

Now that the files and resources are there and the project is loaded into Microsoft Visual Studio .NET 2002, please double check a couple of settings. These settings are all related to the references associated with the project (see *Section 3: Dependencies*).

Reference Settings:

AxPEGOALib	CopyLocal = True
AxPESGOALib	CopyLocal = True
MapWinGIS	CopyLocal = False
MapWinInterfaces	CopyLocal = False
PEGOALib	CopyLocal = True
PESGOALib	CopyLocal = True
stdole	CopyLocal = False
System	CopyLocal = False
System.Data	CopyLocal = False
System.Design	CopyLocal = False
System.Drawing	CopyLocal = False
System.Windows.Forms	CopyLocal = False
System.XML	CopyLocal = False

Now that these settings have been set correctly, click the Build icon, or select Build from the menu. The mwWellViewer.dll has now been created with Microsoft Visual Studio .NET 2002. It is created in the mwWellViewer subdirectory in the /Plugins folder. Next time that MapWindow is run, if the mwWellViewer.dll was built to the correct folder, the updated changes to the Well Log Data Viewer will be available.

Technical Documentation:
Course Resolution Water Quality Model

Last Revision: Dec 28, 2007

Table of Contents

1. *Table Designs*
   - *Table Descriptions and Notes*
   - *Table Relationship Diagrams*
2. *Dependencies*
   - *Software*
   - *Module and Components*
   - *Integrated Development Environment (IDE)*
3. *Setup*
   - *Setup*
   - *Customizations*
4. *Building*
   - *Code Compiling*
   - *Installations*

1. Table Designs

Land Cover Database: LandCoverSummary.mdb
(See "Technical Documentation For Land Cover Summarizer" for table designs)

Best Management Practices Database: BestManagementPractices.mdb
(See "Technical Documentation For Best Management Practices Tool" for table designs)

WRIA-Wide Water Quality Parameters Database: WRIA-WideWQParameters.mdb

Table: Catchments

Field Name	Data Type	Description
CatchmentID	Long Integer	The catchment id associated with the catchment-drainage pair
DrainageID	Long Integer	The WRIA 1 Drainage ID associated with the catchment-drainage pair

Table: EMCs

Field Name	Data Type	Description
DrainageID	Long Integer	WRIA 1 Drainage ID
Land_Cover_Code	Long Integer	The associated land cover class id
EMC_TN	Text	Expected mean concentration (EMC) for total nitrogen (mg/L)
EMC_NH3	Text	EMC for ammonia (mg/L)
EMC_N03	Text	EMC for nitrate (mg/L)
EMC_TP	Text	EMC for total phosphorus (mg/L)
EMC_FC	Text	EMC for fecal coliform (mg/L)
EMC_BOD	Text	EMC for BOD (mg/L)

Table: Input Air Temperatures

Field Name	Data Type	Description
Date	Date / Time	The date the temperature data was recorded
AvgTemperature	Double	Average daily temperature at Abbotsford Canada in degrees C
AirTempFlag	Text	Flag for average daily air temperatures from Abbotsford data

Table: Parameters

Field Name	Data Type	Description
DrainageID	Long Integer	WRIA 1 Drainage ID
DrainageName	Text	Name of the WRIA 1 Drainage
DOPercentSat	Double	DO Percent Saturation Parameter (Calibration Parameter)
AirWaterSlope	Double	Slope of Air Temperature Surface Soil Temperature Relationship (Calibration Parameter)
AirWaterInt	Double	Intercept of Air Temperature Surface Soil Temperature Relationship (Calibration Parameter)
QbTemp	Double	Simulation Base Flow Temperature Deg. C
Cb_TN	Double	Base Flow Total Nitrogen Concentration (mg/L)
Cb_NH3	Double	Base Flow Ammonia Concentration (mg/L)
Cb_NO3	Double	Base Flow Nitrate Concentration (mg/L)
Cb_TP	Double	Base Flow Total Phosphorus Concentration (mg/L)
Cb_BOD	Double	Base Flow BOD Concentration (mg/L)
Cb_FC	Double	Base Flow Fecal Coliform Bacteria Concentration (#/100 mL)

2. Dependencies

The Course Resolution Water Quality Model requires the following software to be installed:

Software:

MapWindow 3.1

DSS Model Manager

Integrated Development Environment (IDE):

Visual Studio .NET 2003 Complete Install

The Course Resolution Water Quality Model requires the data output by the following DSS elements:

Land Cover Summary Tool

Best Management Practices Tool

3. Setup

The WRIA-1_DSS Installation installs the Course Resolution Water Quality Model in the MapWindow Plugins folder.
(This is usually located at C:\Program Files\MapWindow\Plugins\ModelManager\Elements”).

4. Building

To compile the Best Management Practices Tool, add the files below to a Microsoft Visual Studio .NET 2003 Visual Basic .NET Library Project.

mwCourseResolutionWQModel project files:

File Name	Purpose
AssemblyInfo.vb	Contains information relating to the DLL assembly. Generated by VB.NET.
clsWRIAWideWQModel.vb	Runs the course resolution water quality model given the settings provided.
DBClient.vb	Contains the routines used to connect to a database.
DirectoryPicker.vb	Custom component which allows a user to select a directory from the computer's file system.
frmConfiguration.vb	Contains the routines used to select a database.
frmInputs.vb	Allows the user to select the location of input and output data for the model.
frmSelectModelDirectory.vb	A form containing a DirectoryPicker component which allows the user to select a directory from the file system, or a directory to be created by the Water Quantity Model in the current DSS Scenario Run.
frmSelectScenario.vb	Allows the user to select a DSS Scenario Run Id from a list of previously run DSS Scenarios.
WRIAWideWaterQualityModel.vb	Implements MapWindow Plugin and DSS Interface routines, allowing the Course Resolution Model to act as a MapWindow Plugin and a DSS element.

Add the following references to the project:

- DssIntfcLib.dll
- MapWinInterfaces.dll
- mwBestManagementPractice.dll
- System.dll
- System.Data.dll
- System.Drawing.dll
- System.Windows.Forms.dll
- System.Xml.dll

You are now ready to compile the project by clicking the Build Solution menu option under the menu Build.

WRIA-wide Coarse Waterbody Response Model (WWCM) project files

File Name	Purpose
rqn.f95	WRIA-wide Coarse Waterbody Response Model Fortran 95 source file
rqn.bat `lf95 rqn.f95 src\apifunctions.obj -exe "r.exe" -ml lf95 "wqlibs_c.imp"`	Fortran compiler script for creating WWCM executable – executable is r.exe
apifunctions.f95	Library of file management routines through Windows API
	Conversion of LWWLM output to binary format for Fortran model use
am.bat See below	Fortran compiler script for creating WWCM executable
qsetup_n.f95	Manager of model execution for a single time step
setscn_n.f95	Scenario setup routine – executed for each time step
nrmodules.f95	set of numerical routines
reachstructures.f95	coordinates watershed structure definition
condensedll.f95	
stringfunctions.f95	set of string manipulation routines
datesubmod.f95	set of date substitution routines
utils.f95	set of miscellaneous file access routines
wqlibs_c.dll	Dynamic link library holding all of the above routines
cnd.exe	Converts NodeResultsvwxyz.txt files to NodeTSzzz.txt
cnodestots.f95	Source code for cnd.exe
cnd.bat `lf95 cnodestots.f95 src\apifunctions.obj src\nrmodules.obj src\reachstructures.obj src\utils.obj src\datesubmod.obj -exe cnd.exe`	Fortran compiler script for creating cnd.exe

The rp.exe file is built using the Lahey Professional Fortran compiler version 5.6. It has no additional dependencies.

The Course Resolution Water Quality Model is installed in all versions of the WRIA-1 DSS installation.

Automake.fig – Lahey compiler script for WWCM

```
NOQUITONERROR
DEBUG

FILES=.\src\apifunctions.f95
COMPILE=@lf95  %fi -C -O1 -LST -DLL -W -g -trace -nvsw -ml lf95 -ml bc -O %SD%SF%OE -
MOD %mo -TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\utils.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\setscn_n.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\qsetup_n.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\nrmodules.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\reachstructures.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND
```

```
FILES=.\src\condensedll.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\StringFunctions.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

AND

FILES=.\src\DateSubMod.f95
COMPILE=@lf95  %fi -C -o1 -LST -DLL -W -g -trace -ml lf95 -nvsw -O %SD%SF%OE -MOD %mo
-TPP
OBJDIR=.\src\
OBJEXT=obj
MODULE=.\src

LINK=@lf95 @\nooksackmodel\wqlink_c5n.rsp -lst -o1 -exe %ex  -nvsw -dll -ml lf95
Qual2EU_lf95.imp
TARGET=c:\NooksackModel\WQLibs_c.dll
```