

PHP Code Design

PHP is a server-side, open-source, HTML-embedded scripting language used to drive many of the world's most popular web sites. All major web servers support PHP enabling normal HTML pages to embed code fragments that get interpreted before being served up to requesting users. PHP is often used with MySQL, a popular open-source database engine.

PHP is a modern language for web site development with both procedural and object-oriented programming and integrated database support. It often serves as the glue code to connect web pages to a server-side database. An interactive web site is event driven. Each user click, menu selection or keyboard action can initiate a thread of execution that runs many lines of code.

Many web projects start as small code snippets gathered from open source sites. As the web site grows to accommodate new data interfaces, web pages and more user features, new developers are added to the team. Without design and documentation tools, the project often becomes unmanageable, unreliable, insecure and costly to expand.

This paper shows techniques used to organize and communicate the structure of existing PHP projects and design new enhancements. Software models provide the framework that enables a development team to work together productively and understand, evaluate integrate or implement new source code for a PHP project.

- PHP supports procedural programming where each root module calls functions that in turn call other functions creating a thread of execution that can be represented with a hierarchical Structure Chart diagram.
- PHP supports object-oriented programming with classes, attributes and functions. The static structure of object-oriented programs can be represented with a Class diagram using the Unified Modeling Language (UML) notation.
- The data structure of a relational database can be represented with a Data Model diagram, also called an Entity-Relation diagram.

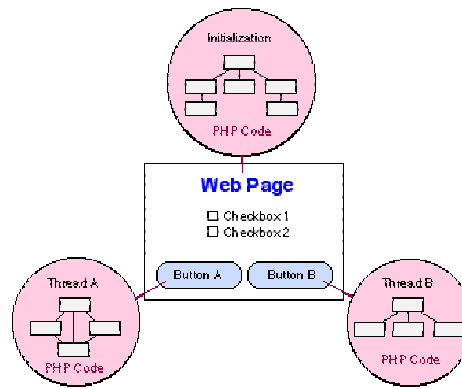
This paper assumes a working knowledge of PHP. It illustrates object-oriented program design with UML class diagrams and procedural program design with structure charts. Tools can automate the process of generating design information from existing code and generating new code from the design.

Modeling Basics

MacA&D and WinA&D from Excel Software are software-modeling tools enriched with PHP language specific details used to generate source code. MacTranslator and WinTranslator are reverse engineering tools that scan code to extract design information to a text file. That information yields structure charts and class diagrams within the modeling tool.

Models can be drawn from a palette of tools. As objects are placed on the diagram, they are given a name and other properties. Each object, be it a module (function), class or database entity has an associated entry in the global data dictionary for the project. Language specific details like a function's parameter list can be entered for PHP or other programming languages. MacA&D and WinA&D use this information to generate PHP code from the model with a simple button click.

PHP programs tend to be event driven by end-user actions like clicking a button on a web page. An HTML form often presents a user interface where the action script runs the root module (function) of some PHP code that calls an inverted hierarchical tree of modules as illustrated below.

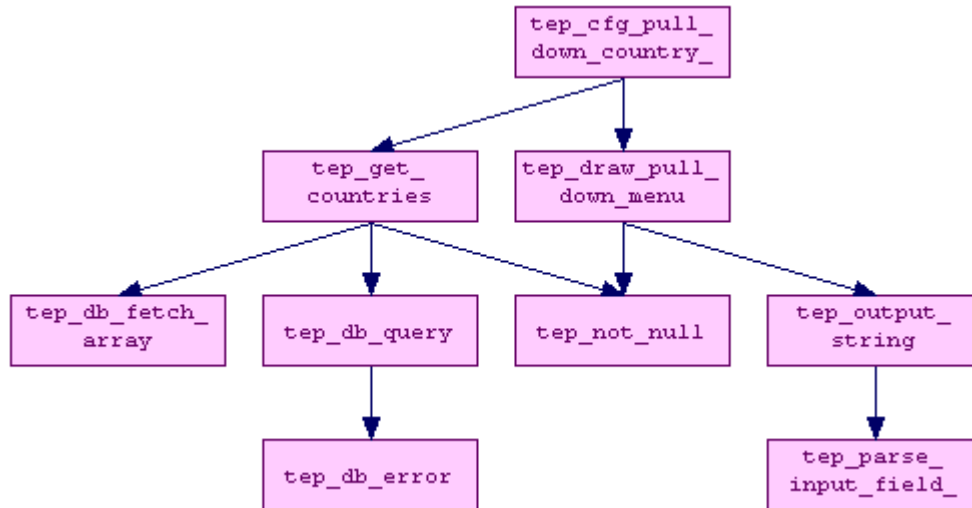


Server-Side PHP Threads of Execution for a Web Page

As web sites grow, they become large software projects with many threads of executions. The structure of each thread can involve a few PHP functions or sometimes hundreds of functions. Programmers can use Structure Chart diagrams to illustrate module hierarchies to understand, communicate, manage and enhance complex web sites.

Structure Charts of Procedural Code

Each thread of execution is represented with a Structure Chart diagram. At the top of the diagram, the root module is shown as a named box. Each called module is shown as a named box connected with a line and arrowhead pointing to it.



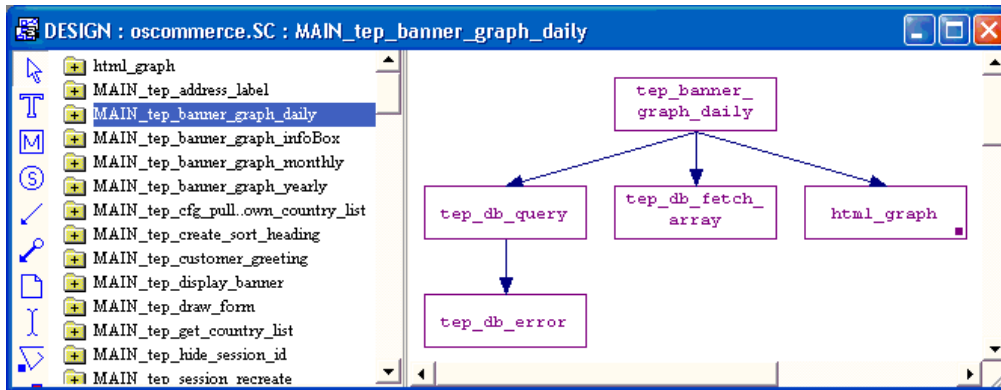
Structure Chart for PHP Thread of Execution

In the open source commerce project illustrated below, WinTranslator was used to process PHP code. The output files are imported into WinA&D to automatically generate structure charts.

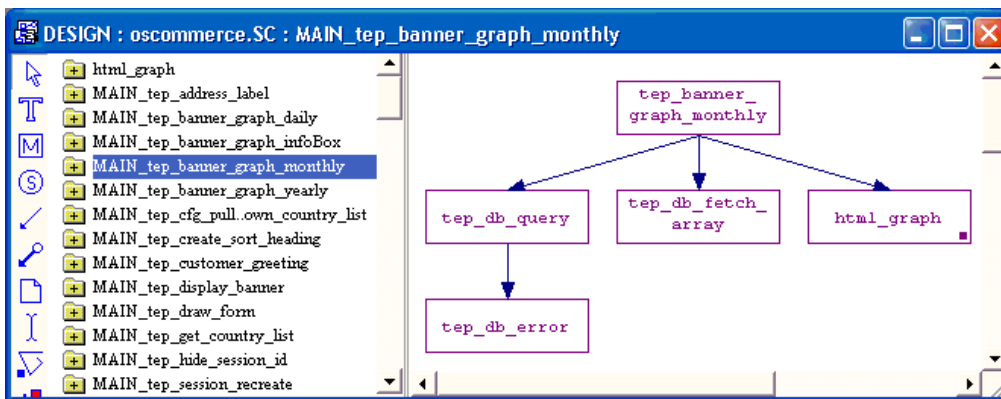
WinA&D has a contents view at the left showing the name of each structure chart diagram. The selected diagram is shown at the right. In the screen shots shown below, you can see that the PHP software can draw graphs daily, monthly, yearly, etc.

The structure of the daily and monthly thread is almost identical. Each chart calls a module named `html_graph` with a small box icon in the bottom right corner of the module. This indicates that the module is linked to a child diagram titled `html_graph` containing a shared branch of code. A developer can double-click that module to see the child diagram.

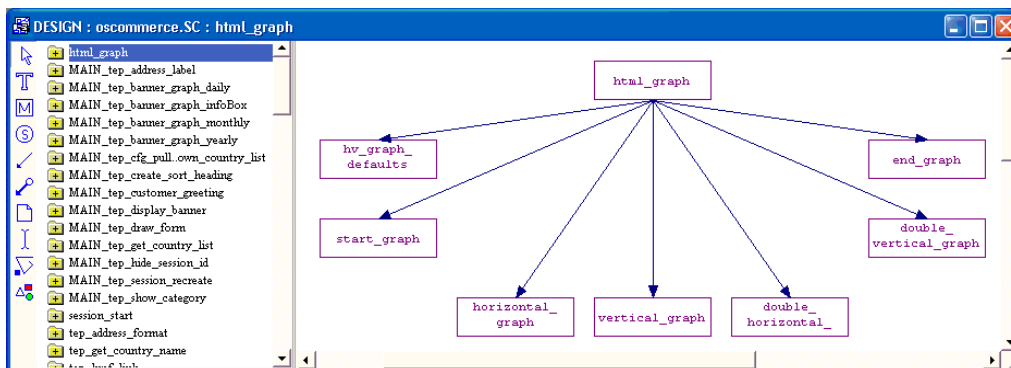
When little human effort, WinTranslator and WinA&D can scan PHP code to automatically document each thread of execution and identify shared branches of code that it places on a separate child diagram.



Structure Chart in WinA&D for Daily Graph Thread



Structure Chart in WinA&D for Monthly Graph Thread



Shared Diagram for HTML Graph

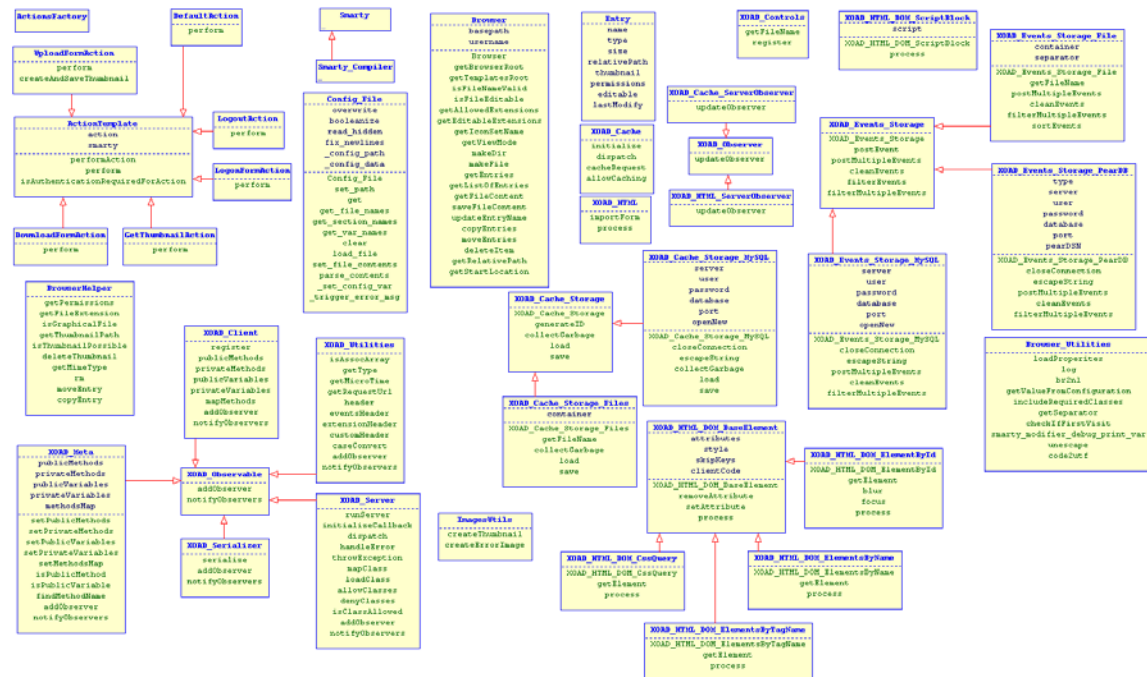
Class Model of Object-Oriented Code

Procedural programs define data structures and functions that access and manipulate that data. As a program grows in size and complexity, the structural hierarchy of its modules also grows. It becomes increasingly difficult to keep track of which functions or threads of execution are manipulating which data. Seemingly small changes can ripple through the software system with unintended affects.

An object-oriented program organizes related data and functions into classes. To reduce the ripple effect of program changes, the actual data structures are often hidden from the other classes or functions that access that data through methods. The physical data structure can change while the interface of the access method and its callers are unaffected. Classes are reusable, self-contained, conceptual units of a program's code and data.

PHP supports object-oriented programming with classes, interfaces, attributes, operations, constructors and destructors. A class can inherit another class, then reuse or override its methods. Attribute and operation members of a class can have public, private and protected access to limit accessibility of class members to other parts of the program.

The static structure of object-oriented programs can be represented with class diagrams using the Unified Modeling Language (UML) notation. UML defines graphical notations for describing and designing object-oriented software systems. It's an open standard controlled by the Object Management Group (OMG). Although UML has many diagram types, class models are highlighted here to show static class structure and relationships.



UML Class Diagram for Online File Browser

A UML class diagram shows each class as a named box. The name of each class attribute member is displayed in one section of the box, with class operation members shown in another section. A class points to the parent class that it inherits.

In WinA&D, a class model can be drawn from a palette of tools. As each class instance is placed on the diagram, it's named in the Class Properties dialog. Each class has a corresponding entry of the same name in the data dictionary where details of that class are stored. Detailed information about a class is stored once, but can be presented on many diagrams. Diagrams within the class model and in other types of diagram documents share data stored in the global dictionary.

For a selected class object on the diagram, the Details button presents the Class Attributes & Operations dialog. This dialog is used to define members of the class. In WinA&D terminology, a class can have Attribute or Operation members. Behind the scenes, WinA&D adds a dictionary entry for each class member with a name of the form Class'Attribute or Class.Operation.

Each class member has a details dialog for defining language specific information for that class member. WinA&D supports many programming languages for code generation including PHP. Depending on which language is currently selected, the Attribute or Operation Details dialog will vary slightly based on specific characteristics of the selected language. WinA&D can concurrently store language specific details for multiple languages for each modeling element.

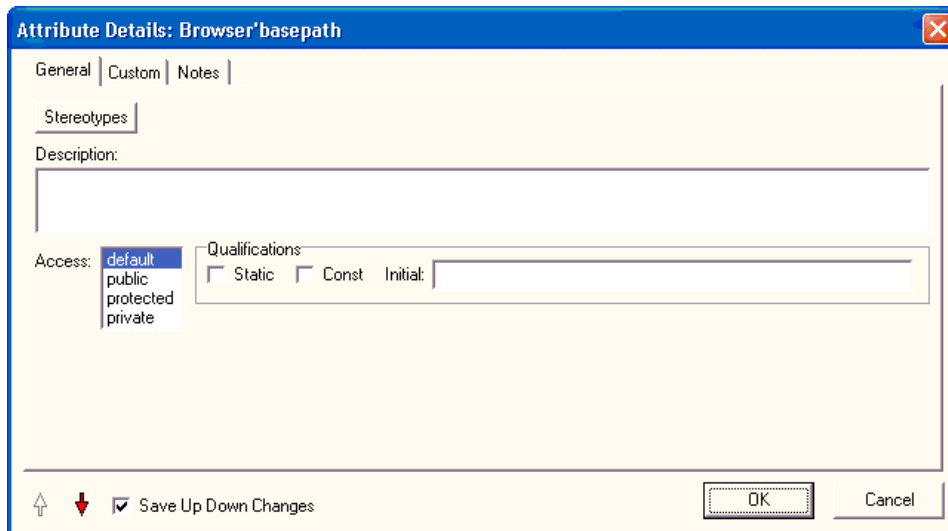
The presentation of a class diagram can vary widely based on user-specified criteria. Indications of public, private or protected access or function parameters can be shown or hidden from the diagram. WinA&D gives the user a lot of flexibility to control how classes are presented. Class members can be shown or hidden based on member type or specific conditions related to the access or modifiers of each class member. Presentation options can be applied across all diagrams, to specific diagrams or to individual instances of a class.

Interface Objects

An interface object defines operation members without implementations. An interface provides a contract that can be implemented by other classes. An interface looks similar to a class box on the diagram with the addition of the <<interface>> stereotype at the top.

PHP Constants & Variables

Constants and variables of a PHP class are represented as class attributes in WinA&D. Details of an attribute like its access, description and qualifications (static, constant and initializer) are entered into the Attribute Details dialog and stored in the dictionary entry for that attribute.

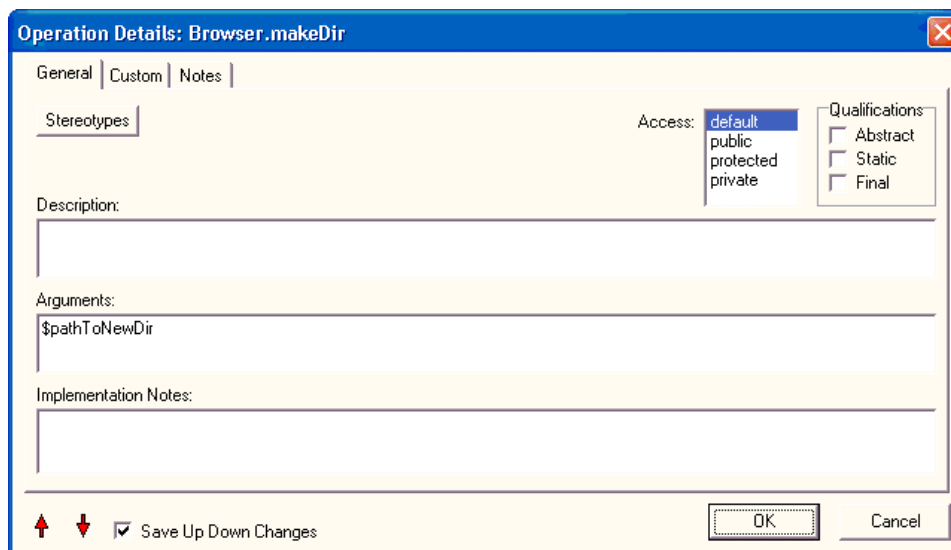


PHP Constants & Variables

The bottom left corner of a class member detail dialog may have Up and Down arrows enabled. Click these arrow icons to navigate between items in a class member list (attributes, operations, etc.) to quickly make editing changes.

PHP Methods, Constructors and Destructors

Methods, constructors and destructors of a PHP class are represented as operations in WinA&D. Details of an operation like its access, description, arguments and qualifications (Abstract, Static, Final) are entered into the Operation Details dialog and stored in the dictionary entry for that operation.



PHP Methods, Constructors and Destructors

A PHP constructor is a method that is executed when an object is instantiated from a class. The constructor name is normally `__construct()`, but for backwards compatibility the method can have the same name as the class itself. PHP classes have an implicit destructor function. Starting with PHP 5.0, an explicit destructor function named `__destruct()` can be defined for a class.

Namespaces

The WinA&D modeling tool has a namespace feature that serves several purposes. Namespaces can partition the models and dictionary information for a large project into different domains like communication, interface, database or control. Many features can be driven by namespace like report generation, import/export and naming conventions. Namespaces are also used to identify the path to code folders that contain the source code for a project.

For Java code, namespaces are used to represent packages during the design phase of a project. For the C# language, namespaces represent that language's namespace construct.

PHP code files are typically stored in one or more code folders. When capturing the design structure of PHP code, namespaces can automatically be assigned to generate class and function entries. These namespaces are then used to organize the generated diagrams and link diagram objects to the associated code files.

Code Generation

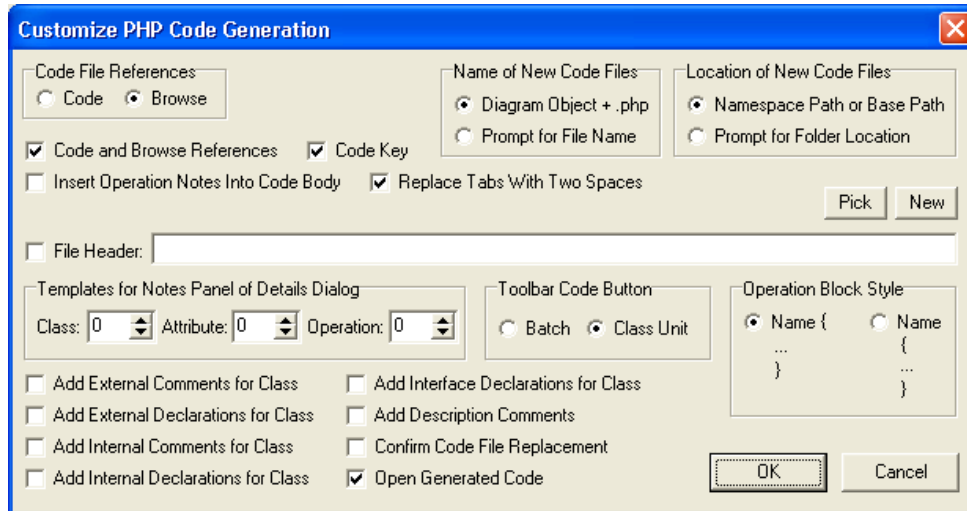
PHP code generation in WinA&D uses information from a class model and its associated dictionary entries. The code generation process is similar to that used for other object-oriented and procedural languages.

For an object-oriented language, the resulting code includes a declaration for each class, or interface with empty function frames for each method. Textual descriptions in the design can be added as class, attribute and method comments in the code.

A checkbox on the customization dialog allows text from the Notes panel of the Operation Details dialog to be inserted into the generated function frame, thus making it easy to include programming comments or source code from the design model into the generated function frames of the PHP code.

Customize Code

The Generate->Code->Customize command presents a dialog to customize the generated code.



Dialog to Customize Generated PHP Code

The Customize dialog provides some control over what gets generated. For example, you can include a predefined file header at the beginning of each generated code file and automatically insert fields like the current time stamp, user name, organization, etc.

Generate Class and Interface Code from Class Diagram

WinA&D has two commands for generating code from a class model, the Batch and Class Unit commands on the Generate->Code submenu.



The Generate Code button on the tool bar runs the Batch or Class Unit command.

The Class Unit command for a selected class A automatically generates the code file named A.php and prompts for a folder to store the resulting file. The Batch code command generates code for one or more selected class or interface objects and prompts for a file name and location. The customization dialog has options that control how a file is named, where its stored and whether the user is prompted for a file name or location.


Generate Function Code from Structure Chart

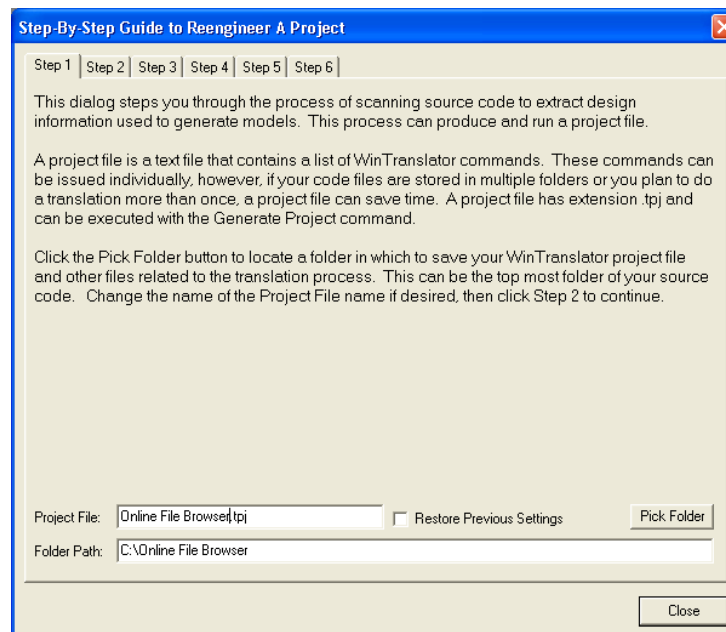
WinA&D can generate a PHP code file that contains selected modules or all modules on a diagram to a new code file or inserted into an existing code file. The generate code frames include arguments, descriptions and implementation notes from the design.

Reverse Engineering

WinTranslator is a reverse engineering tool that scans source code and extracts design information to a text file. That file is imported into empty project files created in WinA&D where diagrams can be automatically generated. The reengineering process is fully automated to quickly produce accurate diagrams of unfamiliar code.

WinTranslator can be used to generate class models from object-oriented code, structure charts from procedural code or data models from SQL. For this discuss, we'll focus on the generation of class models and structure charts from PHP code.

 The user clicks the Reengineer Project button to present a step-by-step dialog that identifies the source language (like PHP), source code folders and other options. WinTranslator scans the code and outputs a text file of design information.



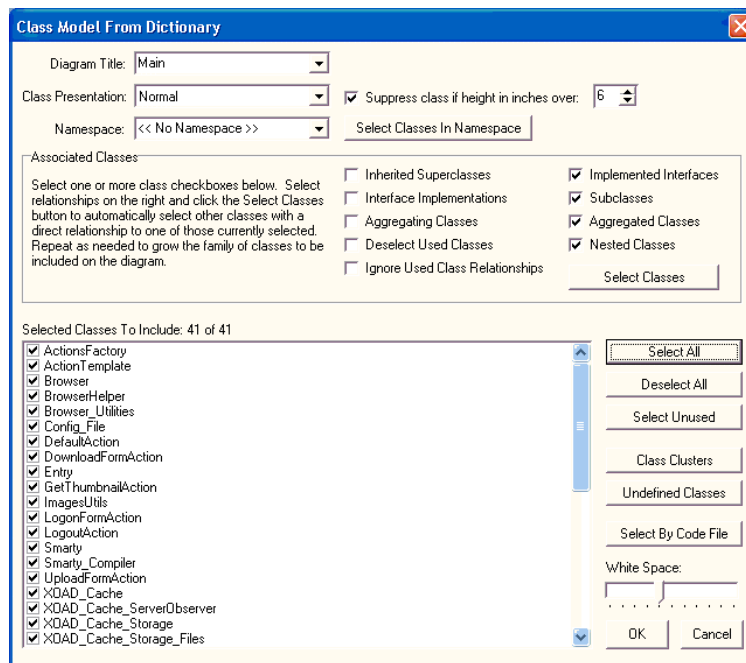
Scan PHP Code to Extract Design Information with WinTranslator

The output from WinTranslator for a PHP project consists of a Dictionary.rp file in each code folder. A Dictionary.list file in the designated WinTranslator project folder references those .rp files.

Within WinA&D, use the New Project dialog to create a new project with Dictionary, Class Model and Structure Model documents and set the language to PHP. Open the generated project documents.

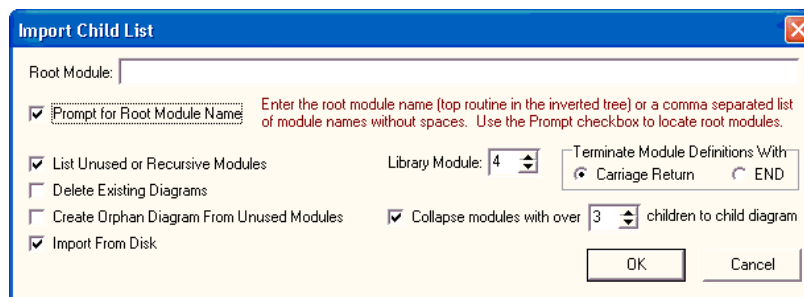
From the Dictionary window, import the Dictionary.list file generated by WinTranslator. The project now has a dictionary populated with design information extracted from the source code. WinA&D has options to color dictionary entries based on type or to structure entries hierarchically to more easily identify entries for classes, interfaces, class attribute and operation members and modules (functions).

From the Class window, use the Generate->Class Model->From Dictionary command to present the Class Model From Dictionary dialog. From this dialog, a class diagram can be generated for each logically related cluster of classes. For a smaller project, all classes can be included on one diagram.



Generate Class Model from Dictionary Information

WinTranslator also produces a Child List.rp file in each code folder and a combined list file named Child List.list in the project folder. That file can be imported into the Design window in WinA&D to generate structure charts.



Generate Structure Charts in WinA&D from PHP Code using WinTranslator

WinA&D has the ability to generate a separate Structure Chart for each thread of execution from the imported child list information. It can also help a developer locate the topmost root module for each Structure Chart diagram.

The Prompt for Root Module Name checkbox will analyze the depth of the module-calling tree under each module to identify candidate root modules. Select one or more modules, and a structure chart will be generated as a separate diagram for each. Diagrams are automatically leveled so modules with lots of child modules get pushed to a child diagram. Several threads of execution may reuse the same branch of a program that is shown on a shared diagram.

Modules on a Structure Chart diagram or class objects on a Class diagram are linked directly to the associated PHP code. A developer can click through diagrams to see the program structure and then click directly to the associated function or class code linked to a diagram object. Code is presented in the integrated Code or Browse window.

Summary

This paper shows how PHP programs can be represented as UML class models and structure charts. Each thread of execution in a procedural PHP program yields a separate structure chart stored in one or more documents. Models can be used as a basis for new code development or generated from existing code to understand its structure.



Excel Software

Ph. 505-771-3719

info@excelsoftware.com

www.excelsoftware.com