Best Practices for Workgroup BIM using Vectorworks

Author: Robert Anderson, Vice President Integrated Practice, Nemetschek Vectorworks
Principal Investigator: Juan Carlos Fabbiani, Senior Project Architect, Nemetschek Vectorworks
Version and Date: 1.0, 9/27/13

Nemetschek Vectorworks
7150 Riverwood Drive
Columbia, MD 21046 USA

T 410-290-5114
F 410-290-8050

www.vectorworks.net

Introduction

Many people ask Nemetschek Vectorworks, "What are best practices for doing Building Information Modeling (BIM) using Vectorworks?" Also, we are frequently requested to set out best-practices recommendations for Workgroup Referencing (WGR) to enable project teamwork. We have undertaken a project to answer these questions in some detail, and an outline and findings of that project are described in this paper.

Audience for this Paper

This "white paper" is not intended to be basic Vectorworks training. The ideal audience for this paper is/are experienced Vectorworks users who have a solid understanding of the basic Vectorworks concepts of:

- Attributes;
- Layers;
- Stories;
- Classes (and Class Attributes);
- Worksheets;
- Resources; and
- Hybrid Symbols;

and who are further well acquainted with the use of the architectural tools in Vectorworks Architect:

- Walls;
- Doors;
- Windows;
- Slabs;
- Roofs;
- Stairs;
- Columns; and
- Spaces;

What do we mean by “Best Practice?”

Wikipedia describes "Best Practice" in the following way:

> *"A best practice is a method or technique that has consistently shown results superior to those achieved with other means, and that is used as a benchmark. In addition, a "best" practice can evolve to become better as improvements are discovered."*

This concept of "evolving benchmark" is a good one, and hopefully will apply in this case. While we feel that the methods outlined in this paper apply to the project described using the current version of Vectorworks as of this writing (Vectorworks 2013 sp4), we are confident this standard will change over time, as the features of Vectorworks are continually enhanced.

At the present time, we feel the best practice recommendations should meet the following broad standards:

1. **Use of BIM Methods:** All Sheets (for presentation and production) and Worksheet Reports must be generated from "live views" onto a single hybrid Building Model. There should be no "breaking of the model" required to do production drawings.
2. **Optimized File Granularity:** There should be enough files in the project so that a realistically large team can simultaneously get work done, but not so many files that they impose significant additional management cost.
3. **Proper CAD Management:** While the methods and the outcomes of "CAD" and "BIM" may be different, there are nonetheless valuable management techniques developed for CAD that still have validity in a BIM workflow. We will identify some of these techniques that apply directly to Workgroup issues, however, this white paper should **not** be considered a comprehensive "CAD Manager's Handbook".
4. **Maximal Efficient Use of Vectorworks Features:** Many users over the years have developed their own workflows. (Some users don't even use automated Door and Window Schedules!) Our project is intended to broadly test the architectural features of Vectorworks as it currently ships, with all features being tested in a multi-user, WGR setting.
5. **All Work-arounds must meet "sanity check":** Many work-arounds are developed by users incrementally and, in the end, add complexity that offsets their utility. All work-arounds should be critically examined to verify that any trade-offs they introduce to the project are in fact a net "win".

Brief Overview of Architectural Test Project

Project type

For our project we decided on a multi-family residential project (see Figure 1). We felt a low- to mid-rise "infill" project comprising mixed use would be both familiar and appropriate. We developed models and drawings for a 12-unit apartment building with commercial/retail use on the ground floor and a developed roof terrace amenity making up in effect a 5th level.

Figure 1 – "White Card" rendering of Architectural Test Project

Project Characteristics

This type of project was chosen for the following primary reasons:

1. **Scale:** This size and type of project is very typical in the small offices using Vectorworks in the USA (comprising 3 to 10 professionals).
2. **Scope:** This kind of project (Multi-family residential infill) could be found in virtually any of the global markets that Vectorworks serves. While the building codes and the national drawing standards may differ, the fundamental approach to "putting the project together" will be the same.
3. **Repetition:** This project involves repetition, not only of units within a floor but also of "common floors". A common requirement of projects like this is the ability to use technology to "leverage" design decisions. Repetition in a CAD context is pretty straightforward (use of instancing) but in a BIM sense is more difficult. It was important to confront problems that might arise from this issue.

Organization of this Document

This white paper presents findings and recommendations in two major sections. The "Best Practices" section will set out a series of general recommendations to follow for management of group work in Vectorworks. The "Project Setup" section will go into some detail about how to organize the various project standard, model and sheet files.

Best Practices

BIM Methodology

What do we mean by "BIM" after all? For our definition of "BIM" (or, as it is sometimes called, "Little BIM"), we require that all drawing output sheets and worksheet reports must be generated from viewports or reports that reference the project 3D / hybrid model. This means that changes to the model will result in automatic changes to all "downstream" views and reports. For most users, we feel this is the “Definition” of BIM in a practical sense.

For Vectorworks users, "BIM" means the ***use of viewports and sheet layers for all sheet output***. While it is technically possible to output plan views from Vectorworks design layers, the facilities provided by the sheet layer / viewport system, including improved viewport navigation, line and text scaling, class overrides, and the Issue Manager, practically limit the practice of output from design layers to casual check prints.

Vectorworks Classes and Class Management

Meaning of "Class" in BIM: Many CAD users are used to thinking about Classes for a single purpose, and that is control of graphic attributes (particularly line-weights and dash styles) in their drawings. This is a capability of AutoCAD layers, and is a typical CAD functionality. However, with certain exceptions (see below), this is not a hard-and-fast requirement of a BIM project, and the degree to which attribute classes are used is up to the user. In BIM, we find it more useful to use Classes to define (i.e. to categorize) "what" something is, and Layers to define "where" (or in whose domain) something is.

Class Standardization: Standard classes makes the Vectorworks project go more smoothly. All model files in the Project should have the same (or a very similar) class structure. We recommend as a starting point the "VWArch" or "AIA_Architectural" templates found in the "Standards" section of the Default Content Libraries. These are accessed any time the user makes a new class, and the user may make multiple classes at once, to populate the entire class list. These class standards may be part of the PRL (see below) file.

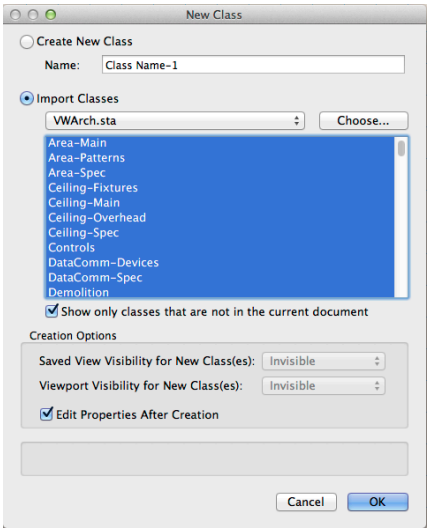

Figure 2 – Creating Multiple Classes from a Template (.STA) file

Class Attributes and Class Overrides: Whether to use drawing with Class Attributes is the decision of the firm's CAD manager. Generally, the use of Class Attributes adds some management control (and complexity) and reduces the "WYSIWYG" nature of the workflow. Class Overrides in Viewports require that the objects in the viewports being overridden be drawn using Class Attributes. Depending on the office graphic standards, we recommend drawing with Class Attributes for the following:

- **Schematic Design (non-component) walls:** This is to allow changing the walls to "solid fill" for plan view schematic presentations. If you don't use this method of display, this is not necessary.;
- **Components of walls used in Design Development / Construction Documents:** This is to allow different presentations of walls in different types of plan views (e.g., changing of lineweights / fills for wall components in

details). If you don't require this level of detail in your plan representation (e.g. if you simply want to hide all or any wall components,) then you may do this with viewport settings and don't require Class Attributes.

All Users Should Reference a "Project Resources Library"

We recommend centralized referencing and control of Resources in a project. This allows obvious graphic standardization on a firm as well as a project basis. The following items should be considered for inclusion in a "Project Resources Library" (PRL):

- Wall-,and Slab-Styles
- Linestyles
- 2D Hatches
- Render Textures
- Graphic Standard Symbols (Title blocks, Space Tags, Section markers, etc.)
- Content Symbols (Doors, Windows, Furniture, Appliances, etc.)

Good CAD management practice suggests that "Version Control" be practiced with this file. That is, an "Office Master PRL" should be maintained in agreement with the current office graphic standards and "branched" with each new project. "Branching" means that a project-specific copy of the Office Master PRL should be made at the beginning of the project (the "Project Master PRL").

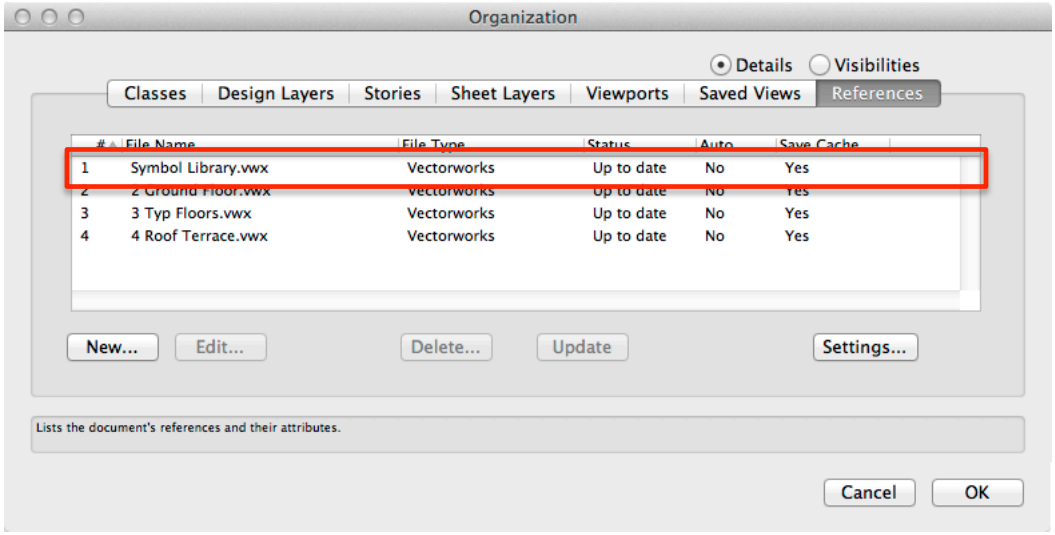

Figure 3 – the PRL ("Symbol Library") should always be the highest priority reference

The PRL when referenced into project files should always be given the highest priority reference to all users. This avoids resource naming conflicts when referencing files that themselves have referenced resources.

(Physical) File Sharing Issues

File Sharing: In all but the very smallest offices, a dedicated file server is recommended. The incremental cost these days of a gigabit switched network (over a slower one) is non-existent, so set up the fastest infrastructure possible. All files should be directly opened from the server and not "checked out" to local machines, unless WAN-type distances (e.g. in work-from-home situations) are in effect.

Autosave: We strongly recommend using Autosave; pick the frequency settings you find most useful. To avoid interruptions during auto-saves, set the backup path to your local hard drive (see Figure 4).

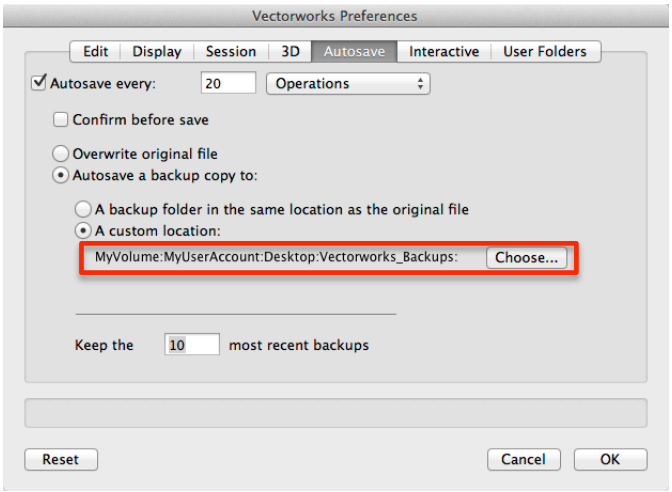

Figure 4 - Set the Autosave path to your local Hard Drive

DropBox: Although it gives the casual appearance of one, Dropbox is not a file server. It is a very sophisticated file synchronization service, but it cannot detect when files have been opened by the user. Therefore, for project file service, it is not a viable solution.

Project Folder Setup

We recommend setting up a project folder with a /Models folder. The PRL file should be put at the root level of this folder, and each project phase should have a subdirectory. Within each phase subdirectory, there should be a Model files subdirectory, with the A-### Sheet files individually placed (see Figure 5). All references in a project should be relative path based, to provide for project portability (moving from one physical location to another) as well as to allow for project branching and versioning as necessary.

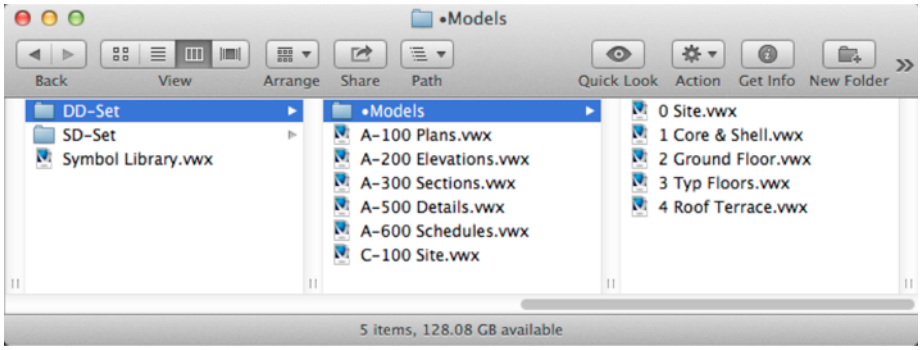

Figure 5 – A typical project folder structure.

Door, Window Management Issues

In our reference project we chose to have all doors and windows entered as symbols (each containing a standard Door or Window plug-in object). We felt this was mandated by the repetitive nature of the building fenestration. However, this in turn mandates that Doors and Windows be scheduled by Type, and not by Instance. There are two rules that must be followed for the use of symbols-of-PIOs to be successful:

- The door and window symbols must be created from doors and windows inserted in walls, and not just sitting free in the document; and
- Each combination of door-in-wall and window-in-wall must be represented by a symbol. (In other words, a 3070 slab door with a particular hardware set is a different symbol if it is going into an interior framed wall and an exterior composite masonry wall.)

Users who need instance-based scheduling (which would more likely be the case for Doors rather than Windows) must use plug-in objects, and not symbols, for Doors and/or Windows as appropriate.

At this time, the Door Hardware Library.TXT file located at *[Vectorworks folder]/Plug-ins/VW_Arch/Data/Prefs_Def/* must be managed manually for the architectural workgroup. We recommend that the office specifications expert use Vectorworks to set up a master "Door Hardware Library.TXT" file and distribute it to all users to replace the file in the location above.

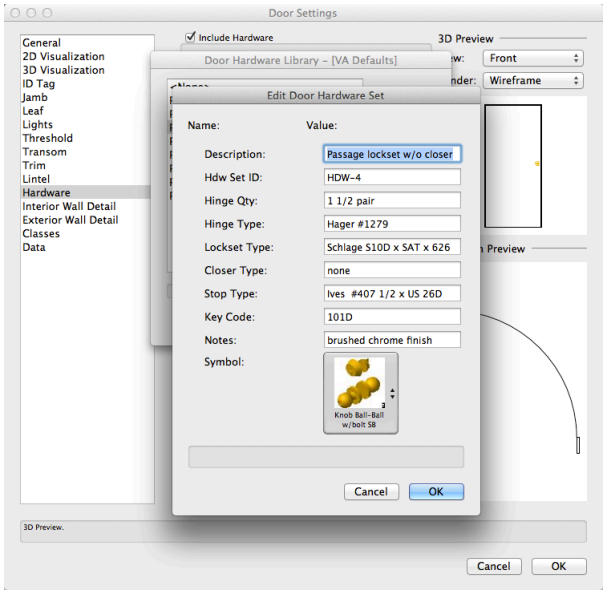

Figure 6 – Managing Door Hardware information

Understanding Workgroup Referencing

Vectorworks allows two different kinds of Workgroup Referencing: Layer-based, or "classic" style, and design-layer-viewport-based (DLVP-based), or "modern" style. Each of these types have advantages and disadvantages, so it is useful to understand them in a little more detail.

"Classic" Layer referencing

Layer-based referencing can be can be thought of as a massive automated "Cut and Paste" of everything that is on the referenced layer of an external document. Almost all Vectorworks systems work transparently, but this referencing style can be "messy", because all classes used in referenced layers will be brought into the target file, and thus will require class name synchronization. Also, new Vectorworks users may run into trouble if they draw on referenced layers (if they're not paying attention to the alerts that Vectorworks puts up, see Figure 8) because changes they make to referenced layers will be lost upon update.

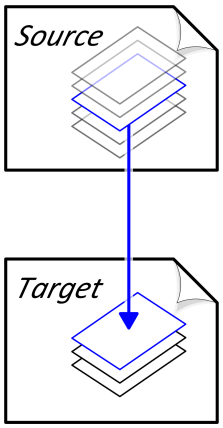

Figure 7 – Layer-referencing

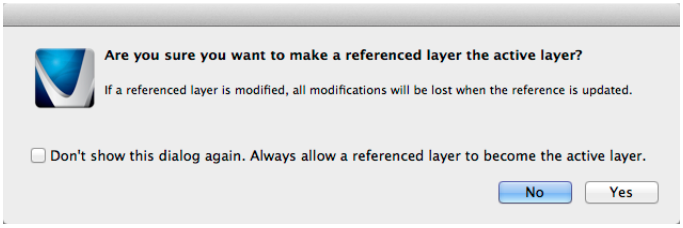

Figure 8 – Warning alert seen when trying to draw on a referenced layer

Layer referencing should be used when the target file will require the user of class overrides, as class overrides are not supported by DLVP references. Also, source information that is on a Story-layer will conform to a similarly named story in the target file, so story bindings are another reason to use "classic" referencing.

Using a layer-reference seems "natural", particularly if the user wants an exposed layer structure (rather than one encapsulated in a DLVP—see below). Using a layer-reference can also create a smaller target file than a DLVP reference, as it brings in only the information contained in the referenced layers.

"Modern" Design Layer Viewport (DLVP) referencing

Design Layer Viewport referencing packages up an entire source file and allows its contents to be selectively viewed (by source-file layer and class) without bringing the layer and class structure of the source file into the target file. As DLVP-referencing brings in the entire source file, it can make the target file size larger (when compared to a layer-referenced target file.)

DLVP-referencing is convenient where views of the entire building are needed (e.g. Building Elevations and Renderings), as they bring views of the entire building in as a single viewport object, requiring only one design layer to contain.

Likewise, DLVP-referencing is convenient when the user is creating reports on the entire building (rather than for a single floor.) This can require some care and oversight when referencing multiple files that themselves have reference layers, to avoid double-counting of model elements.

Lastly, DLVP-referencing does not support class overrides of referenced data, so it is unsuitable for situations where class overrides are desirable (certain styles of floor plan views, for example.)

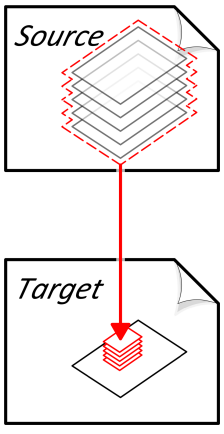

Figure 9 – DLVP-referencing

Two Simple Rules for Referencing

The "Either-Or" Rule: A Vectorworks file may incorporate layer-style ("Classic") referencing, or DLVP ("Modern") referencing, but not both. See the dialog in Figure 10; observe the highlighted controls.

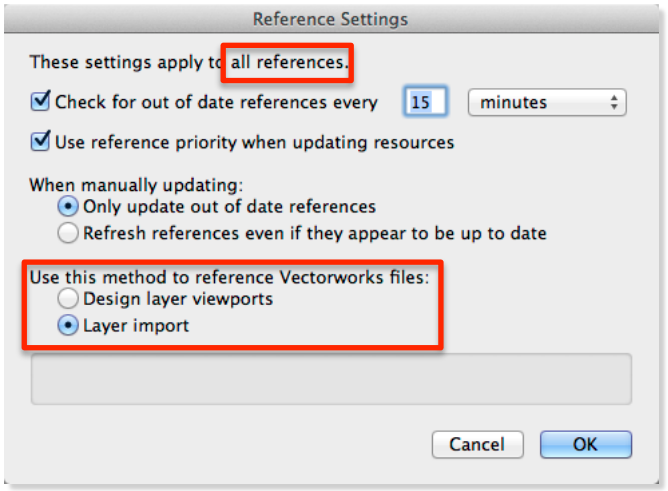

Figure 10 – The Reference Settings dialog

The "No Nested Reference" Rule: Simply stated, this rule is, information that is referenced in a Vectorworks file may not serve reliably as source data in another reference.

Team Size and Project "Granularity"

No matter the office size, teams for a particular project may grow to a certain size and no larger. There is a temptation to break a project up into a lot of files for "maximum flexibility". However, there is a significant additional overhead that many, many files in a project impose.

Most teams start small and experience a late-stage "boom", where people get added to "get the project out the door". In the "charrette", many people are usually employed doing (usually) annotation work—notes and dimensions—on sheets. So there is a temptation to pursue a "one sheet per file" strategy. However, for Issue Management, this imposes additional work, as the files need to be managed independently. The ideal is to provide enough granularity that everyone can stay engaged, but no more, so additional management costs are not incurred.

We recommend the following general approach to project structuring:

- **One "Core & Shell" Model file:** Generally, exterior design is handled by an single individual or a very small group working in a tightly integrated manner. For this reason, and for the streamlining that a single file affords, we recommend a single file for the building exterior (see "General Modeling Approach: Separate the Building Exterior and Interiors" below).
- **One Interior Model file per floor:** The interior construction (non-structural walls, ceilings, fixtures, interior doors) for each floor of the project is handled in a single file, so the project model may be handled a floor at a time by an individual.
- **One "Collaboration" file for "BIG BIM" export:** This file aggregates the model for IFC and other kinds of BIM collaboration.
- **One file per type of output sheet (Site Plan, Plans, Elevations, etc.):** The number of physical files should be minimized to optimize the effort needed for issue management. Sheet files are easily duplicated (and sheet layers added/deleted) to add a team member as needed during "crunch time".

Performance Tips and Tricks

Curved 3D Geometry, Speed, and File Size

Curved 3D geometry (spherical, cylindrical, and conic shapes, NURBS surfaces and 3D edge filleting) can create performance problems. This is because to create a 3D rendering or a 2D projection of a 3D shape, it must "tessellate" (i.e., convert to a triangular mesh) the 3D shapes for the rendering engine. For excessively detailed geometry, this can result in the following three issues:

- Slow 3D rendering times;
- Slow generation of section and elevation viewports;

- Large 3D file export size;

What is meant by "excessively detailed geometry"? Sometimes, curved geometry (particularly very detailed curved geometry such as fillets in door hardware or cabinet pulls) is not essential to the representation, and may effectively be eliminated without losing any significant information in the view. The geometric detail needed for a door handle may be very different when representing a single room versus an entire hospital.

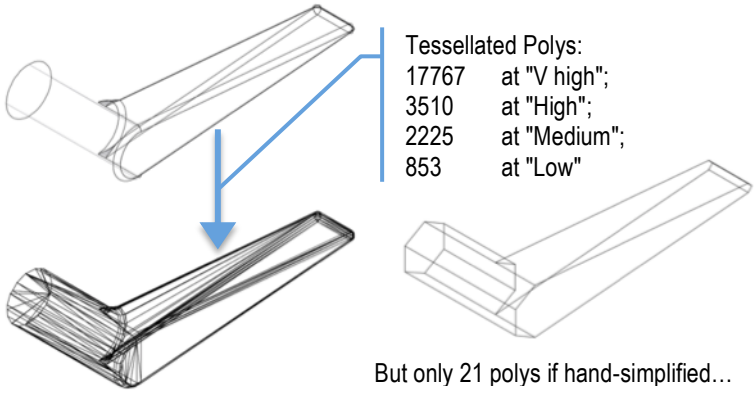

Figure 11 – Hand-simplification can create much simpler shapes than 3D Conversion Resolution

In general, 3D Conversion Resolution can have a very large impact particularly on section viewport generation. It is important to realize that this setting is set on an individual viewport basis (see Figure 12), and that these viewport settings are not affected by the application preference setting:

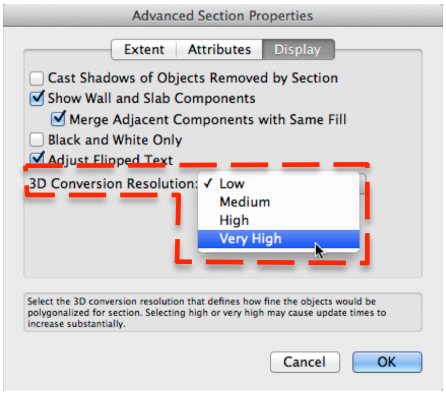

Figure 12 – Set Conversion Resolution for Section Viewports in the Advanced Section Properties dialog

Project Origin

Sometimes (for reasons of alignment with a governmental grid, or interoperability reasons) it is necessary to operate with a highly offset origin. In Vectorworks (as in virtually all other CAD and BIM programs), drawing and modeling objects a very long distance from the coordinate system origin (see Figure 14) will cause performance problems. For this reason, it is strongly recommended to always draw near the coordinate system origin and use an offset user origin to achieve the desired coordinate offsets (see Figure 15). When importing DWG files to create a project background, use the "Center After Import" option to ensure this is done automatically (see Figure 13).

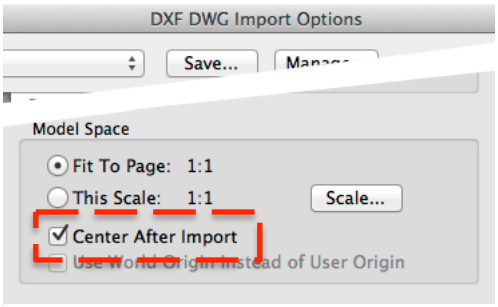

Figure 13 – Center After Import option for DWG

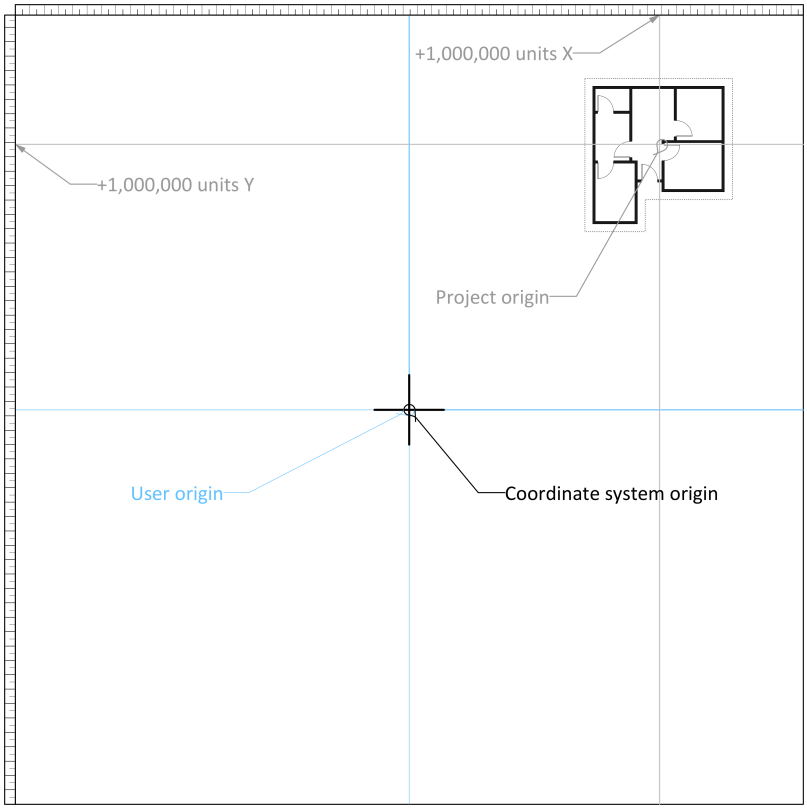

Figure 14 – Incorrect way to handle large-offset project origins

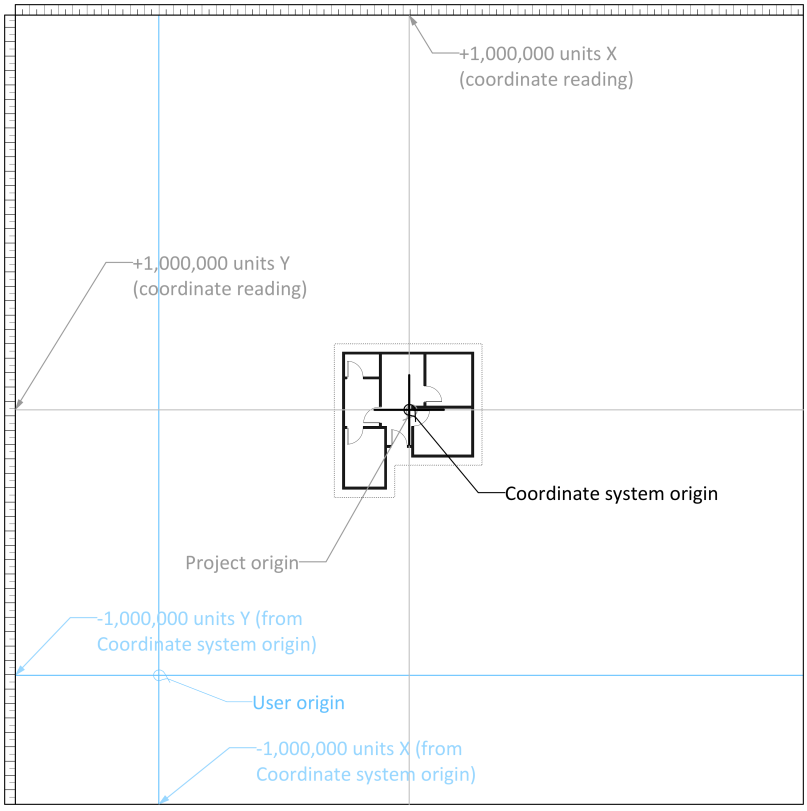

Figure 15 – Correct way to handle large-offset project origins

Project Setup

Conceptual Project Structure of 3 “Tiers”

The "Project Files" (by this we mean files used to produce project images and sheets) can be thought of as existing at one of three different levels: Standards, Model, or Sheet files. While not absolutely hard and fast (for example, some Model files will begin to incorporate project "standards" over the course of the project), these are nonetheless useful as an organizing approach to files:

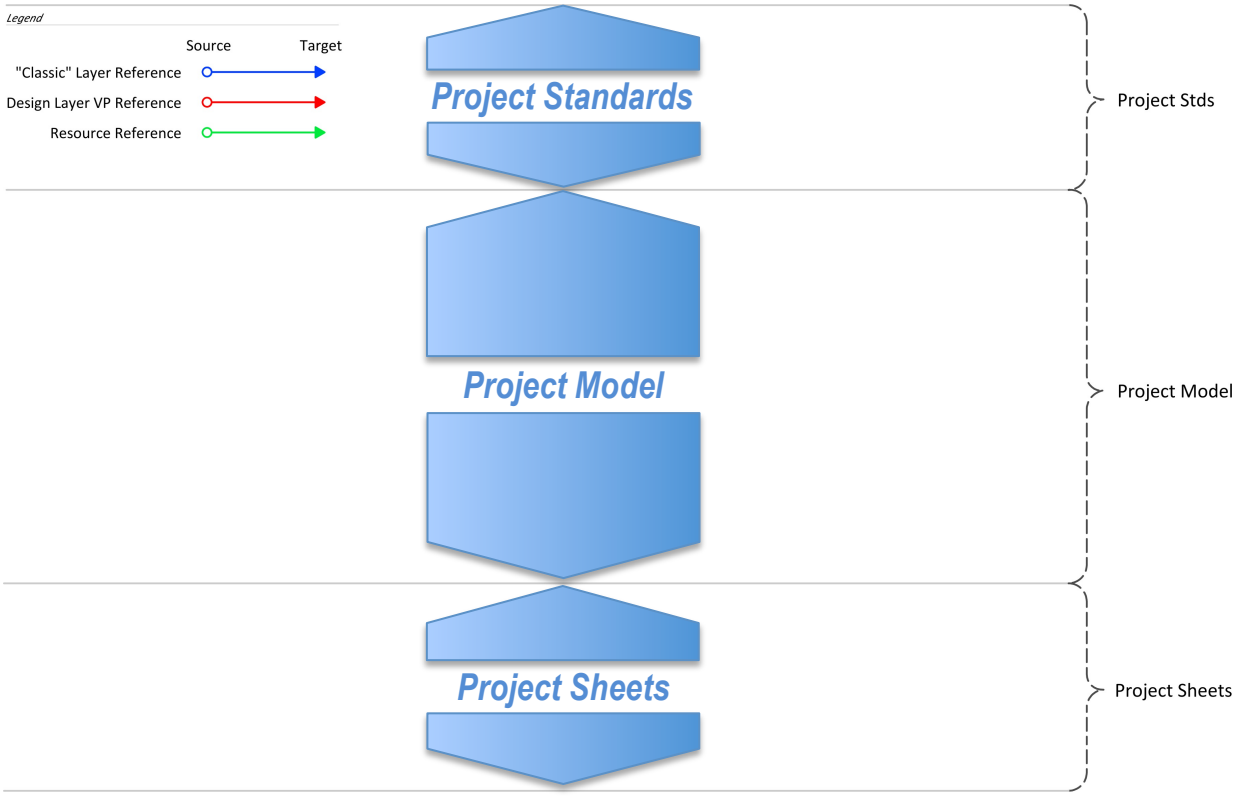

Figure 16 – A project is composed of 3 "Tiers" of Files: Standards, Model, and Sheets

The topmost "tier" is the **Project Standards**, which are used by all parts of the project that follow. Depending on the office, this may be one or several library and/or stationery files, comprising classes, symbols, styles, attributes, and other graphic standards. The middle "tier" is the **Project Model**, a series of files that holds the entire BIM model for the project. At the bottom are the **Project Sheets**, the presentations and graphic output for the client presentations and contract documents.

This tiered diagram will serve as the background for the "Project Referencing Maps" to follow. Note the **legend** at upper left that shows the different types of referencing used in the project.

General Modeling Approach: Separate the Building Exterior and Interiors

In line with our discussion at Team Size and Project “Granularity" above, we propose a file structure for the project that reflects the structure of most small- to medium-sized office project teams. This structure involves separating the building exterior from the project interiors. The project exterior file describes the main structure and exterior of the building for all floors. This file contains:

- Exterior Walls, Doors, Windows, Balconies, other Features
- Slabs and Roofs
- Columns, Beams and Structural Walls
- Stairwells and Shafts
- Foundations and Basements

This "Core and Shell" (C+S) file is organized into Stories, and all design layers are story layers. Our reasoning in making the entire building exterior a single file is simply that, in most offices, a single senior architect will control the project exteriors and structure-to-envelope detailing. In any case, splitting the project exterior into multiple files imposes a significant coordination cost and complexity that (no matter the project team size) offsets any possible gains.

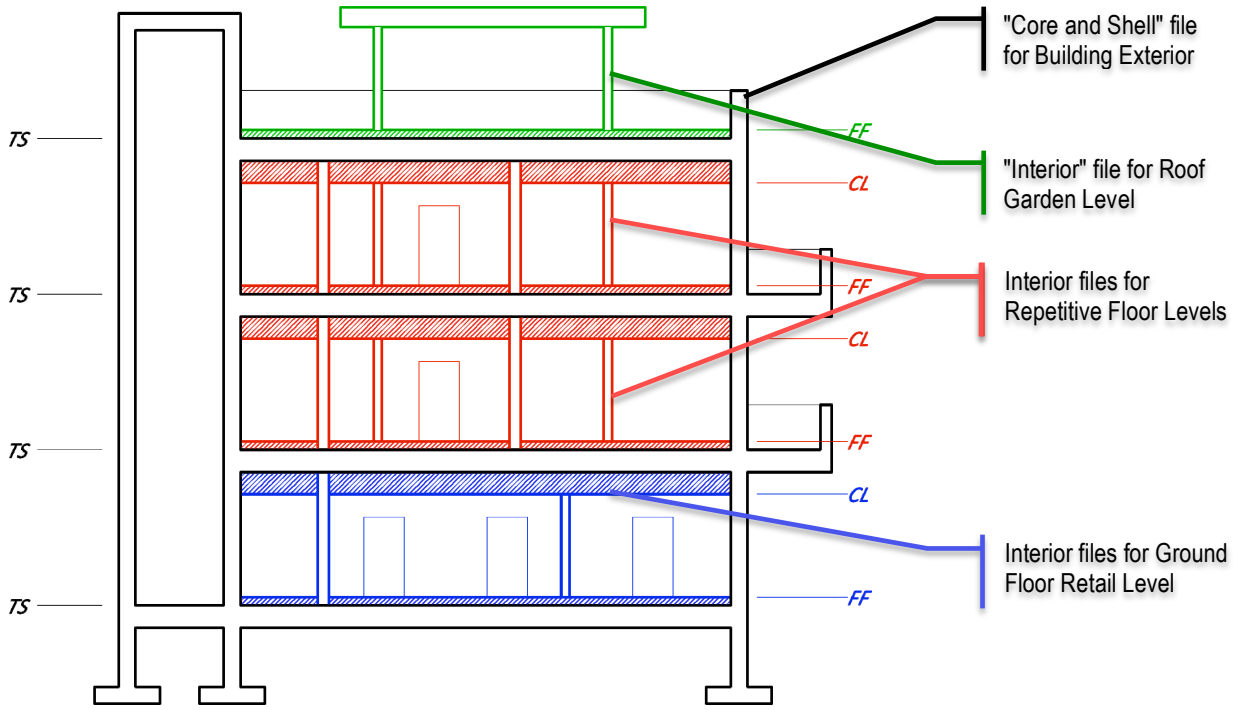

Figure 17 – The Model Files Separate the Building into Interior and Exterior Elements

The project interior files, of which there are several, represent the interior construction for each level, including:

- Demising walls and corridor walls
- Interior partitions, doors, and windows
- Spaces and finishes
- Millwork and fixtures
- Ceilings, lighting, furr-downs

Each of these interior construction files are set up with individual stories whose names match the stories in the C+S file. In addition, the Level Types used by their layers may not conflict with (i.e. duplicate) Level Types used in the C+S file.

Project Reference Map—Project Resource Library

The next sections of this paper will sequentially build up the Project Reference Map for our multi-family residential project.

The Project Resource Library (PRL), referenced by virtually all project files, contains not only symbols, but all resources: Line styles, Wall and Slab styles, Tile fills, Title blocks, etc.

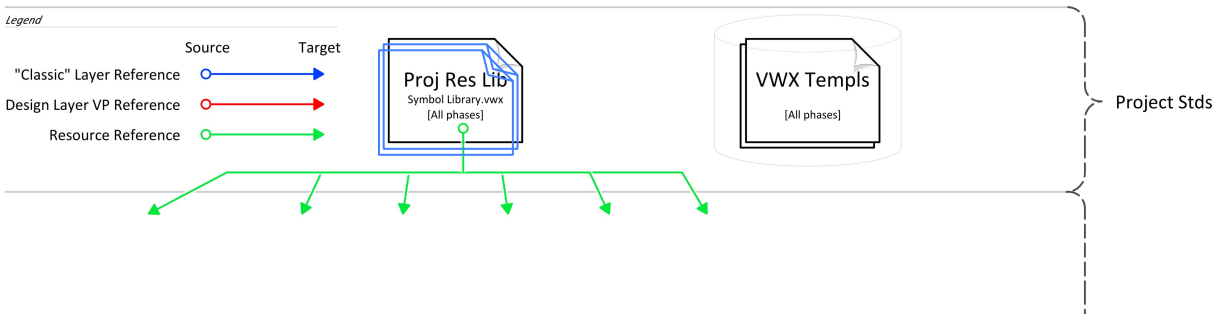

Figure 18 – The Project Resource Library files

Once taken from the ongoing office standard templates (AKA "branched from" the office standards,) it is project-specific and should not be synchronized with the office standards except for a very good reason. Some offices will have this file be several files, but there is really no reason for it not to be a single file.

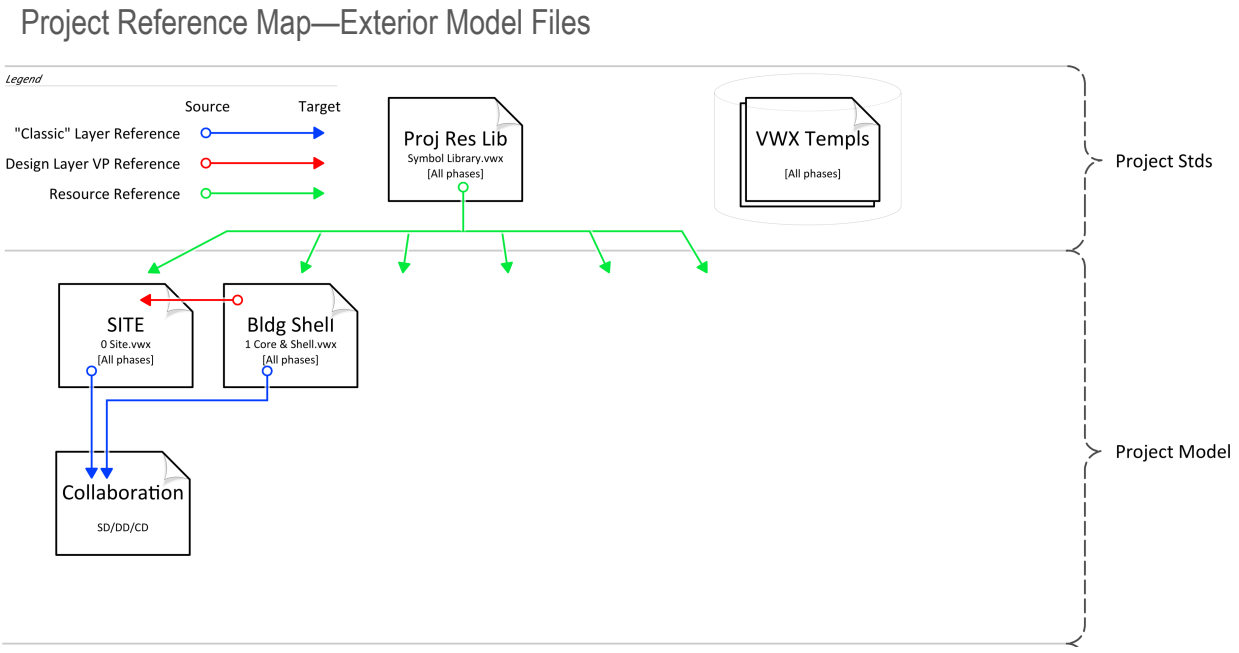

Figure 19 – The Exterior Model files

The **Exterior Model File** has been previously described in some detail. This file (called here the "Building Shell" file as an alternate to "Core and Shell") is DLVP-referenced by the **Site File** (which contains the Site Model, Site Modifiers, paving, and other site improvements) to allow accurate creation of site modifiers based on the building / foundation. The **Collaboration** file is used for export to IFC and/or gbXML over the course of the project as necessary. Since both IFC and gbXML file formats requires stories, it is set up for stories; most easily it is made from a template based on the Building Shell file.

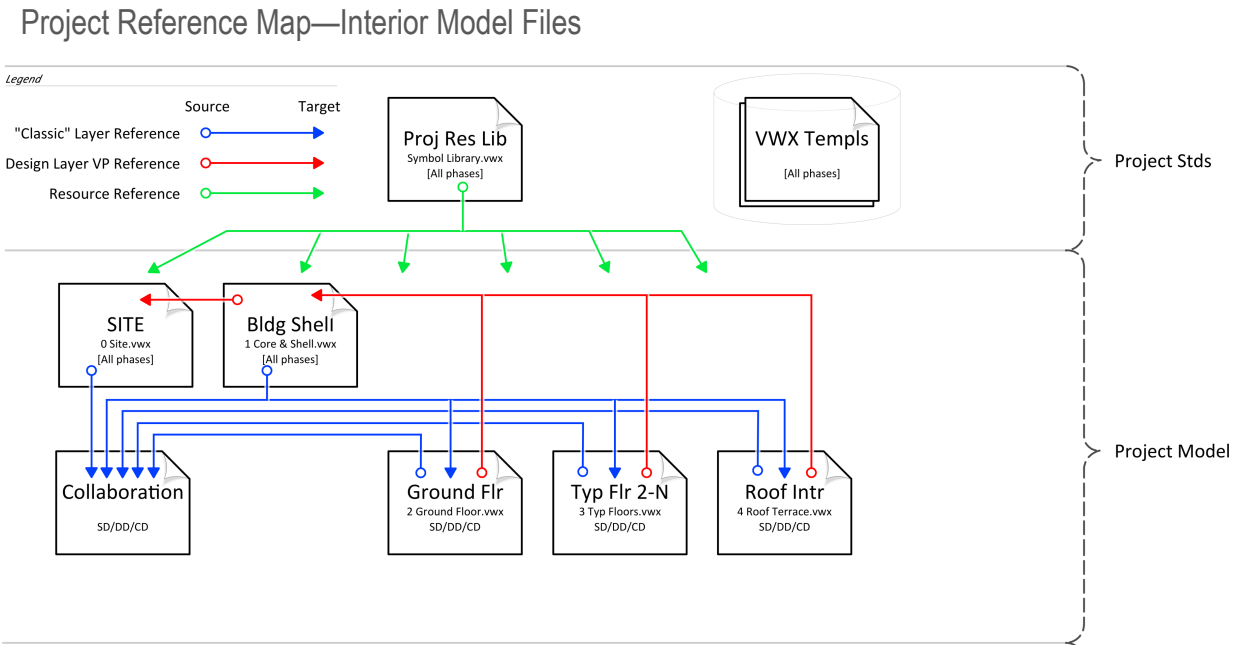

Figure 20 – The Interior Model files

The **Project Interior Model Files** have been previously described in some detail. The interior files may be DLVP referenced back to the shell file for temporary coordination **only**. (In this case, the DLVP references require dedicated design layer that can

be hidden to avoid circular referencing.) As previously mentioned, the Interior Model Files are set up for Stories with Story-names matching the Building Shell file. The "Typical Floor" file (containing multiple identical floors) is set up for multiple stories. In this file, the units are symbols containing all the elements of the unit. This means that the units must have similarly named and numbered spaces and doors.

Project Reference Map—Schematic Design Presentation

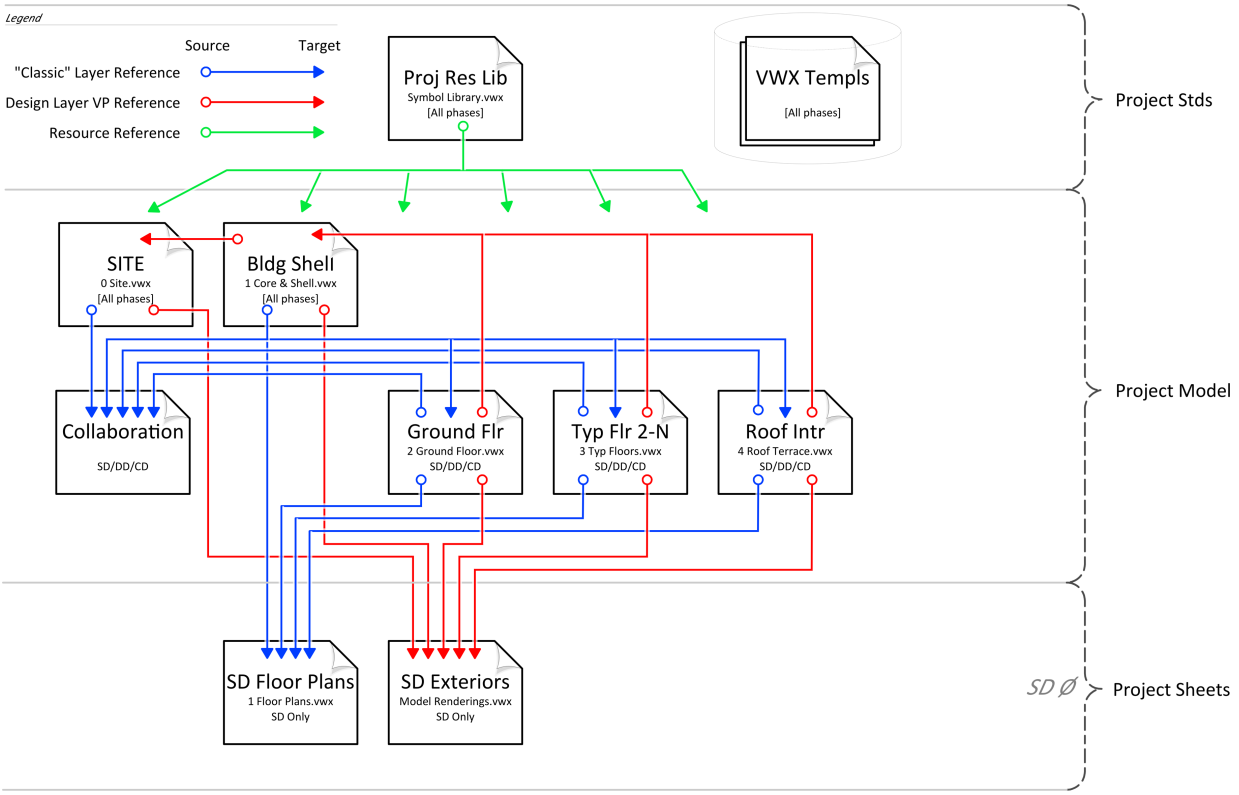

Figure 21 – The Schematic Design Project, including Sheet Files

The sheets in the Schematic Design phase use both **Layer Referencing** (for the plan sheets) and **DLVP-referencing** (for the exterior elevations, building cross-sections, perspectives, etc. The Layer Referencing is necessitated because floor plans require class overrides. The sheets in the "SD Exteriors" file are never presented on a floor-by-floor basis, and so may use the simpler-to-manage DLVP referencing for elevation, section, and perspective drawings.

Archiving by Phase: At the end of the SD phase, the entire project should be archived, with all the model and sheet files. The user should duplicate the entire set of Schematic phase model and sheet files (maintaining the folders) and lock them for record keeping. In the new "development" folder the model will change (e.g. components for walls and slabs).

Project Reference Map—Design Development / Construction Docs

In the Design Development and Construction Documents phases, the types of sheet output are diversified. This can be most easily done simply by duplicating and renaming files and reworking the sheet layers on each file.

We propose as best practice putting section lines and elevations markers on **dedicated design layers within their respective files**. These layers get layer-referenced back to the plan sheets to provide “live” section lines in files separate from the actual section files. See the references with the Section and Elevation sheet files as their sources in Figure 22. Wherever DLVPs are used, as in the Elevations file, the DLVPs must be locked in place in these files to maintain proper registration for the section/elevation markers.

Since so much notation and dimensioning work gets applied to floor plans, clearly this is one of the main opportunities for additional “granularity” for offices that want this, and so it is possible to have more than one set of plan sheets. They could be duplicated and broken out between ceiling and floor plans, for example, or by floor. But since each sheet file has to be managed independently with regards to Issue Manager, this imposes a management cost.

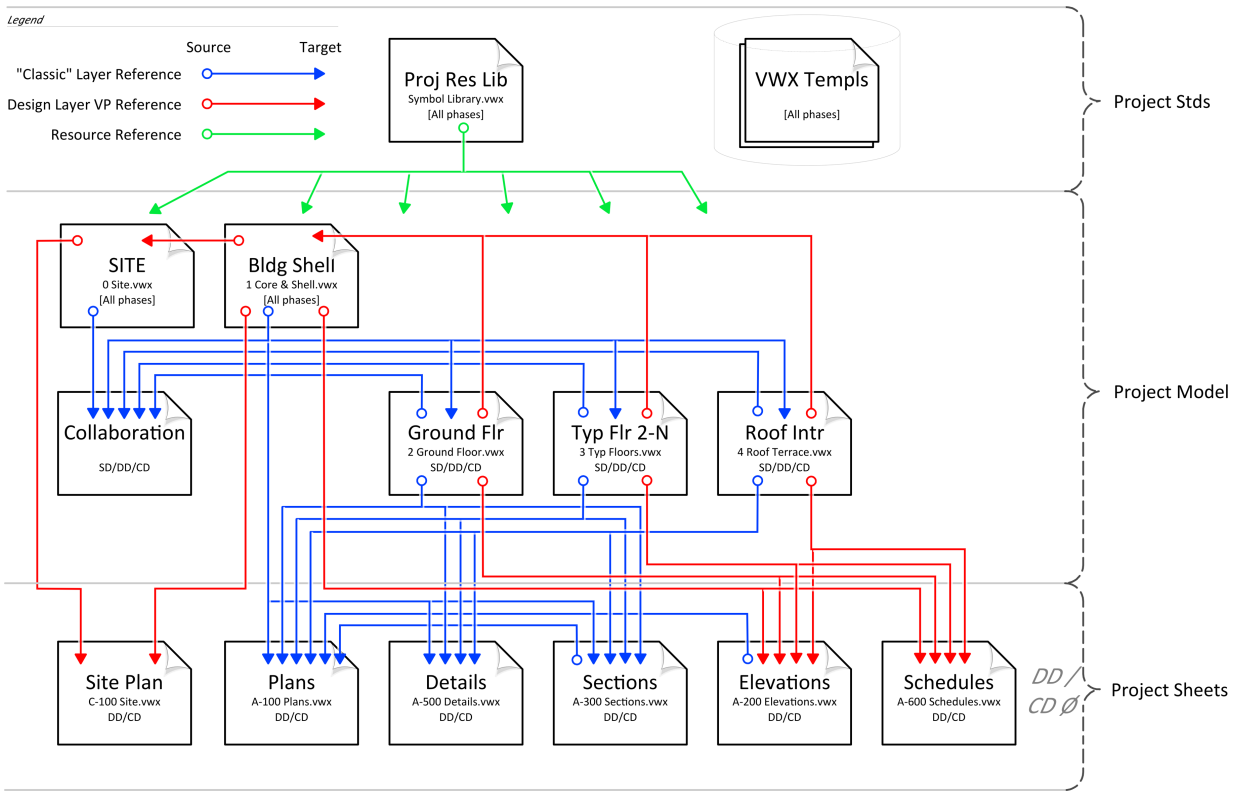

Figure 22 – The Project Setup for Design Development / Construction Documents

Conclusion

The project working files for this project may be found on the "BIM in Practice" pages on the Vectorworks website. This paper has not attempted to be a comprehensive "CAD Manual" for the use of Vectorworks. Nonetheless, we have tried to explore and resolve many issues facing the Vectorworks practitioner who is involved in an integrated, multi-user BIM project. We believe the workflow presented depicts a realistic and practical approach to this management challenge. We welcome any feedback on the recommendations and positions put forth herein.

5 July, 2013

Robert Anderson, randerson@vectorworks.net
Juan Carlos Fabbiani, jfabbiani@vectorworks.net