Case Study:
Zend Server on IBM i

Vermont Gas Systems
Work Order Management System
About John Valance

- **Independent consultant**
  - Specialty is helping iSeries shops develop web applications, and related skills
  - Training, mentoring, consultation and coding
- **25+ years IBM midrange development experience**
- **12+ years of web development experience**
  - Web scripting language of choice = PHP
- **Frequent presenter on web development topics**
- **Trainer for Zend Technologies**
  - Teaches Intro to PHP for RPG programmers
  - Zend Certified Engineer
- **Contact:** [johnv@jvalance.com](mailto:johnv@jvalance.com) - 802-355-4024
Agenda

• Project Background
  ▸ Company
  ▸ IBM i / PHP
  ▸ Project Goals

• Demo of System
  ▸ recorded

• Technical Details
  ▸ major architectural features
Vermont Gas Systems

- Natural Gas Utility in north western Vermont
- Regulated Business
- Serves Burlington and surrounding areas
- About 40,000 customers (small utility)
- Expanding service territory
IBM i (aka: iSeries, System i, i5, AS/400)

- IBM's legacy midrange platform
- Precursors date back to 1970s -80s
- Unique design, legendary reliability and longevity
  - High technology investment protection
- Proprietary, integrated, object-based operating system (i/OS)
  - Many built-in business capabilities
    - Database (DB2), Security, Communications,
  - Technology Independent Machine Interface
    - OS has survived many hardware technology changes
- Proprietary programming language = RPG
- Vast portfolio of 3rd party business applications, in all industries
  - Typically character-based terminal applications (aka green-screen)
- Runs enterprise applications, backbone for many medium to large businesses
Screen-shot: IBM i Sign On Display
PHP on IBM i

- 2005: Zend/IBM partnership
  - Zend/PHP = Strategic technology for IBM i
  - IBM i = Strategic platform for Zend
- Simple Installation includes Zend Server CE, licenses for Zend Studio
- PHP widely acceptance by IBM i community
  - more accessible for RPG programmers than Java
  - Demand for education is growing
- Typically used to access legacy DB2 tables
- Toolkit for IBM i
  - Access native IBM i system objects
  - Call RPG programs
Background - Reasons for a New System

- Project Scope: Rewrite VGS Work Order Mgmt System in PHP
  - Need to replace 15 y/o legacy, green-screen application, with numerous enhancements
  - Old system needed many enhancements, was difficult to maintain
  - Lots of redundant code, hard-coded value lists

- Technical goals of new system:
  - Modern, intuitive user interface
  - Solid, modular code base
  - Easily maintainable and extensible
Why Custom PHP Solution?

- Search for new system included several vendor offerings
  - One vendor offering was $1 Million+ solution, plus services
  - Excellent solution, but...
    - Would have completely disrupted existing business processes
- In the end, decision to use custom coded solution
  - Browser-based interface
  - PHP / Zend Server running on IBM i
  - Consultant (me) to lead project and write code
Benefits of Custom Approach

• Get exactly what they needed
• Low risk
• Low cost

System functionality based on old system, with enhancements

• Meet pressing business requirements
• Greatly improved interface and code base
• Incremental improvements to business processes
• Allows for future enhancements
• **Maintain Information on Work Orders**
  - Anything related to construction or maintenance of gas lines
    - Gas Transmission lines (regional pipelines)
    - Gas Mains (local pipelines)
    - Gas Services (pipe to premises)
  - Repair Leaks (on mains / services)
  - Retire / Replace main or service
  - Work Order Type = describes type of line and work to be performed

• **Primary users are Engineering Department**
  - Accounting also researches W/O issues
Ancillary Information

- Cleanup details
- Pipe Exposures
- Sewers on site
- DIMP Data Collection (Key project goal)
  - Distribution Integrity Management Plan

Interfaces

- Marketing - Sales Applications (New Installs)
- Accounting Activity / Project Costs
  - Time Sheets / Payroll
  - Vendor Billing
Life Cycle of Work Orders

- Create Work Order
- Print Work Orders
  - Several different formats for each WO type
- Complete Work Orders
  - Entry of data collected in field
- Close Work Orders
  - Post details to accounting
Technical Features of the System

- Hybrid Object Oriented / Procedural Design
- Model / View / Controller code organization
- Selective use of Zend Framework components
- Powerful custom helper classes
  - Form generation
  - CRUD SQL generation
- Highly consistent, easily maintainable code base
  - Table searching/filtering/download
  - Single record CRUD screens
- User maintainable list management (Drop-downs)
- Security
  - Authentication with IBM i UID/PSWD
  - Robust access control at user or group level
- Use of JOD Reports to generate PDF reports and printed forms
Demo of System

VGS Work Order Management System
Main Menu

Work Orders
- All Work Orders
- Leaks
- Create New Work Order
- Create Multiple SR/ST Orders
- Pipe Exposures
- Cleanups
- Sewers
- Plastic Pipe Failures
- Mechanical Fitting Failures

Tables
- Projects
- Pipe Types
- Drop Down Lists

Security
- Profiles
- User/Group Xref
- Authorities
- Profile Authorities
- Test with profile:
  [Enter user name]  [Change user]

Reporting
- Cancelled Work Orders download
- W/O Inventory Reconciliation

System
- Log Out
Use of Zend Framework

- Began with desire to implement Zend Framework based architecture
- Not full implementation of Framework
  - R & D period factored into schedule
  - After 2 months, could not get all features working
    - Zend_DB_Table, Zend_DB_Select, Zend Paginator
  - Created home-grown versions of these components
    - VGS_DB_Table, VGS_DB_Select, VGS Paginator
- Used many of the concepts of Framework, but had to "roll our own" classes
  - Very consistent design, using OO
- No front controller, but using a common layout.php on all pages
  - Handles authentication, error checking, page layout for every screen.
- Extensions to Zend_Form
  - VGS_Form extends Zend_Form
- VGS_Form_Helper
  - Automatically sets many form attributes from DB2 metadata
General Application Screens Structure

- **layout.php** - Included in each view script
  - Functions to show header and footer
  - Error handling: `set_error_handler()`
  - Session management: `session_start()`
    - Check `isset($_SESSION[\'userId\'])`
    - If not, redirect to loginCtrl.php

**Two main types of applications:**

- **Search Screens**
  - Multi-record, filtered, paginated record lists
- **Edit/Display Record screens**
  - Single-record screens, for CRUD operations

Cookie-cutter design for building list and record screens
Search Screens - components

- **Nav Buttons bar**
  - Main menu (or close pop-up)
  - Download (optional)
  - Create record (optional)
  - + Custom buttons (app specific)

- **Filter bar**

- **Paginator bar**

- **Search Results** (records matching search criteria)

- **Download capability**
  - Uses filters entered for search
### Example – Work Order Search

#### VGS Work Order Management System

<table>
<thead>
<tr>
<th>WO#</th>
<th>Description</th>
<th>Type</th>
<th>Status</th>
<th>Municipality</th>
<th>Entry Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>67527</td>
<td>16 Stanley Ct</td>
<td>SI - Service New Construction</td>
<td>Pending</td>
<td>St. Albans City</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67526</td>
<td>15 Stanley Ct</td>
<td>SI - Service New Construction</td>
<td>Pending</td>
<td>St. Albans City</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67525</td>
<td>13 Stanley Ct</td>
<td>SI - Service New Construction</td>
<td>Pending</td>
<td>St. Albans City</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67523</td>
<td>37 Saint Albans Rd</td>
<td>SI - Service New Construction</td>
<td>Pending</td>
<td>Swanton Town</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67517</td>
<td>Starbird Road</td>
<td>MI - Main New Construction</td>
<td>Pending</td>
<td>Jericho</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67516</td>
<td>7 Ducks Ct</td>
<td>SM - Service Maintenance</td>
<td>Pending</td>
<td>Milton Town</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67515</td>
<td>Thorpe Ave. Ext @ Fairfax St.</td>
<td>MM - Main Maintenance</td>
<td>Pending</td>
<td>St. Albans City</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67514</td>
<td>79 Walnut St</td>
<td>SM - Service Maintenance</td>
<td>Pending</td>
<td>St. Albans City</td>
<td>Aug 28, 2013</td>
</tr>
<tr>
<td>67513</td>
<td>31 Diamond St #1</td>
<td>SM - Service Maintenance</td>
<td>Pending</td>
<td>St. Albans City</td>
<td>Aug 27, 2013</td>
</tr>
<tr>
<td>67511</td>
<td>2 Canada St</td>
<td>SM - Service Maintenance</td>
<td>Pending</td>
<td>Swanton Village</td>
<td>Aug 26, 2013</td>
</tr>
<tr>
<td>67510</td>
<td>Wheatley Ct</td>
<td>TI - New Construction Tie-In</td>
<td>Pending</td>
<td>Colchester</td>
<td>Aug 26, 2013</td>
</tr>
<tr>
<td>67509</td>
<td>Wheatley Ct</td>
<td>MI - Main New Construction</td>
<td>Pending</td>
<td>Colchester</td>
<td>Aug 26, 2013</td>
</tr>
<tr>
<td>67508</td>
<td>Woodside Dr for # 92</td>
<td>TI - New Construction Tie-In</td>
<td>Pending</td>
<td>Colchester</td>
<td>Aug 26, 2013</td>
</tr>
</tbody>
</table>
Edit/Display Record screens - components

• **Nav Buttons bar**
  - Main menu (or close pop-up)
  - Save (not in display mode)
  - Cancel
  - Return to Search List
  - + Custom buttons (app specific)

• **Form**
  - Field Groups (defined in Form class)
Record detail screens - CRUD

• **Modes:**
  - Create, Read (display), Update (edit), Delete

• **Create and Edit modes**
  - Same validations, by default
  - If form data changed, warning on cancel or navigate away

• **Display mode**
  - Same form layout as Create/Edit
  - Set all inputs to readonly, class="disabled", tabindex="-1", and clearValidators()

• **Delete mode**
  - Show record output only, format like display mode
  - Replace "Save button" with red "Delete button"
  - Confirm dialog before delete
Three primary objects are involved, related to table being maintained:

- **Form object, extends Zend_Form, e.g.:**
  - class Project_Form extends Zend_form { ... }

- **Form Helper object (custom code, very helpful)**
  - class VGS_FormHelper { ... }
  - Retrieves DB2 metadata for table fields on form
  - Sets appropriate filters, validators, attributes based on metadata

- **Table object, extends VGS_DB_Table, e.g.:**
  - class Project_Master extends VGS_DB_Table { ... }
• HTML forms are unwieldy, and contain many co-dependent elements
  ‣ <form> tag
  ‣ <input> tags of various types
  ‣ Field labels
  ‣ CSS and other rendering attributes
  ‣ Error messages
  ‣ Data values

• Server side code must handle:
  ‣ Form loading (new record defaults/ existing record values)
  ‣ Validations based on data types, business rules
  ‣ Reloading form with values and messages, until valid input
Zend Framework provides classes to simplify/abstract form processing

- Handle all aspects of form definition and processing in PHP
- Use `render()` method to display the form.
- Largely avoids HTML, controls everything via structured program code.

- **Zend_Form** ~= `<form>`
- **Zend_Form_Element** ~= `<input>`
  - Zend_Form object contains multiple Zend_Form_Element objects
$form = new Zend_Form;
$form->setAction('resource/process')
    ->setMethod('post');
$username = new Zend_Form_Element_Text('username');
$username->addValidator('alnum')
    ->addValidator('regex', false, array('/^[a-z]/'))
    ->setRequired(true)
    ->addFilter('StringToLower');
$form->addElement($username);
Zend_Form methods

- **reset()** - clear form or load defaults for new record in create mode
  - can override in derived class, with application appropriate values
- **populate($dataArray)** - load existing record in edit/display/delete mode
  - keys of $dataArray are names of form elements
  - good idea to use DB field names
- **validate()** - perform custom validations on inputs
  - will run all validators added
    - you can create custom, reuseable validators
  - can override in derived class with custom validations
- **render()** - generate HTML <form> and <input> tags for all elements
  - handles all attributes, including value="..."
  - decorators handle positioning (HTML container tags)
abstract class VGS_Form extends Zend_Form { ... }

VGS_Form
- $conn
- $fh
- $inputs
- $mode
- $return_point
- $screen_title
- $valid
- __construct($conn)
- activate() : void
- buildLookup($fieldName) : void
- convertDateFormat($dateStr, $fromFmt, $toFmt, $padLen=8)
- createRecord() : void
- deleteRecord() : void
- fixDateInput(string) : void
- fixDateOutput(string, $blnOutputOnly=false) : void
- fixTimeInput(string) : void
- fixTimeOutput(string) : void
- getTimeStampOutputFormat($timestamp) : void
- isCreateMode() : void
- isDeleteMode() : void
- isInquiryMode() : void
- isUpdateMode() : void
- loadScreen() : void
- preProcessFormInputs() : void
- processScreen() : void
- renderFieldGroup($fieldGroupName) : void
- renderFormButtons() : void
- renderFormHeaderMessage() : void
- renderFormHiddens() : void
- renderFormJS() : void
- renderFormTop() : void
- retrieveRecord() : void
- returnToCaller() : void
- setDateFormat($field) : void
- setInputFormatsForDB2() : void
- setInputFormatsForScreen($data) : void
- updateRecord() : void
- validate() : void
VGS_Form in a Nutshell

- Extends Zend_Form (we get all that goodness!), plus...
- Basic data access methods (create, update, retrieve, delete)
- General DB2 data filtering (screen <-> database)
- Form rendering
  - Field groups, elements, buttons, messages, hidden fields, JavaScripts.
- Form processing
  - Initialization, loading from database, validation, database update, redirecting after success
- Boolean mode methods
  - isInquiryMode(), isUpdateMode(), etc...
- Attaching input helpers / popups
  - Lookups (foreign key table search popup)
  - Date pickers
VGS_Form_Helper

- Loads metadata from DB2 for one or more tables
- Builds form elements from the metadata for selected fields
- Adds appropriate data type validations, filters and attributes, and labels to form elements
  - This saves a lot of tedious coding
- Allows definition of field groups (boxes of fields on screen)
- Fields can be added to form by passing comma-separated list of field names.
- Can define additional validations and attributes for lists of fields
  - Required entry, output only, override input type, attribs
class VGS_FormHelper

VGS_FormHelper
- $elements
- $fieldGroups
- $metaData
- $mode
- $view

- __construct()
- addCustomMetaData($name, $text, $type, $length, $precision="") : void
- addElement($name, $element) : void
- addElementsFromMetaData() : Count
- addFieldGroup($fieldList, $fieldGroup, $caption) : void
- addMetaData($conn, $table) : void
- buildElementFromMetaData($elemMeta) : void
- getElements() : void
- getMetaData() : void
- getObjetLibrary($object, $objType) : void
- renderFieldGroup($fieldGroup, Zend_Form) : void
- setDescription($fieldName, $description) : void
- setElementDataTypeFilter(Zend_Form_Element, $meta) : void
- setElementProperty($name, $property, $value) : void
- setElementsProperties($namesList, $property, $value) : void
- setMultiOptions($fieldName, $optionsList) : void
- splitNames(string) : true
class VGS_FormHelper
{
  /** The $metaData array will be used to generate labels, * filters and validators for the form elements automatically.
   * @var array */
  public $metaData = array();

  /** Contains an array of Zend_Form_Element to include on the form
   * @var array */
  private $elements = array();

  /**
   * Holds an array describing the field groupings for display
   * @var array */
  public $fieldGroups = array();
class WO_SewerForm extends VGS_Form
{
    private $wswRec; // Record array for existing w/o sewer record
    private $woRec; // Complete w/o record for the related w/o

    // Key fields for this sewer record
    private $woNum;
    private $wswSeqNo;

    public function __construct( $conn, $woNum ) {
        parent::__construct ( $conn );

        {constructor stuff}...

        $this->fh->addMetaData($conn, "WO_SEWER");
        $this->setDefaultElements( );
    }
}
public function addMetaData($conn, $table) {
    $schema = self::getObjectLibrary($table, '*FILE');

    $syscols = new VGS_DB_Table($conn);
    $query = "select * from qsys2/syscolumns
        where table_schema = '$schema'
        and system_table_name = '$table' ";
    $rs = $syscols->execListQuery($query);

    while ($sysColumn = db2_fetch_assoc($syscols->stmt)) {
        // Add each column's metadata to the master metadata array
        $colName = $sysColumn['COLUMN_NAME'];
        $this->metaData[$colName] = $sysColumn;
    }
}
### WSW_ADDRESS - MetaData

- **UPPER_CASE** = fields from QSYS2/SYSCOLUMNS  
- **Green** = attributes used to generate form elements  
- **Red** = attributes added by custom code

<table>
<thead>
<tr>
<th>FIELD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_NAME</td>
<td>WSW_ADDRESS</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>WO_SEWER</td>
</tr>
<tr>
<td>TABLE_OWNER</td>
<td>ORCOM</td>
</tr>
<tr>
<td>ORDINAL_POSITION</td>
<td>3</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>VARCHAR</td>
</tr>
<tr>
<td>LENGTH</td>
<td>100</td>
</tr>
<tr>
<td>NUMERIC_SCALE</td>
<td></td>
</tr>
<tr>
<td>IS_NULLABLE</td>
<td>N</td>
</tr>
<tr>
<td>IS_UPDATABLE</td>
<td>Y</td>
</tr>
<tr>
<td>LONG_COMMENT</td>
<td></td>
</tr>
<tr>
<td>HAS_DEFAULT</td>
<td>Y</td>
</tr>
<tr>
<td>COLUMN_HEADING</td>
<td>Address</td>
</tr>
<tr>
<td>STORAGE</td>
<td>102</td>
</tr>
<tr>
<td>NUMERICPRECISION</td>
<td></td>
</tr>
<tr>
<td>CCSID</td>
<td>37</td>
</tr>
<tr>
<td>TABLE_SCHEMA</td>
<td>WORKORDT</td>
</tr>
<tr>
<td>COLUMN_DEFAULT</td>
<td>''</td>
</tr>
<tr>
<td>CHARACTER_MAXIMUM_LENGTH</td>
<td>100</td>
</tr>
<tr>
<td>CHARACTER_OCTET_LENGTH</td>
<td>100</td>
</tr>
<tr>
<td>NUMERIC_PRECISION_RADIX</td>
<td></td>
</tr>
<tr>
<td>DATETIME_PRECISION</td>
<td></td>
</tr>
<tr>
<td>COLUMN_TEXT</td>
<td>Address</td>
</tr>
<tr>
<td>SYSTEM_COLUMN_NAME</td>
<td>WSWADDR</td>
</tr>
<tr>
<td>SYSTEM_TABLE_NAME</td>
<td>WO_SEWER</td>
</tr>
<tr>
<td>SYSTEM_TABLE_SCHEMA</td>
<td>WORKORDT</td>
</tr>
<tr>
<td>USER_DEFINED_TYPE_SCHEMA</td>
<td></td>
</tr>
<tr>
<td>USER_DEFINED_TYPE_NAME</td>
<td></td>
</tr>
<tr>
<td>IS.IDENTITY</td>
<td>NO</td>
</tr>
<tr>
<td>IDENTITY_GENERATION</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_START</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_INCREMENT</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_MINIMUM</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_MAXIMUM</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_CYCLE</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_CACHE</td>
<td></td>
</tr>
<tr>
<td>IDENTITY_ORDER</td>
<td></td>
</tr>
<tr>
<td>group</td>
<td>sewer</td>
</tr>
<tr>
<td>include</td>
<td>1</td>
</tr>
<tr>
<td>label-class</td>
<td>required</td>
</tr>
<tr>
<td>required</td>
<td>1</td>
</tr>
</tbody>
</table>
public function setDefaultElements( ) {

    $this->fh->addFieldGroup($flWO, 'wo', 'Work Order Details');
    $this->fh->setElementsProperties($flWO, 'output_only', true);

    $flSewer =
    'WSW_SEQNO, WSW_ADDRESS, WSW_CITY, WSW_LOCATED_PRIOR, WSW_SEWER_SIZE,'
    'WSW_SEWER_MATERIAL, WSW_SEWER_TYPE, WSW_SEPARATION_FROM_GAS,
    WSW_INSPECTION_NEEDED, WSW_INSPECT_REASON';

    $this->fh->addFieldGroup($flSewer, 'sewer', 'Sewer Information');

    $this->fh->setElementsProperties(
        'WSW_SEQNO',
        'output_only',true);

    $this->fh->setElementsProperties(
        'WSW_ADDRESS, WSW_CITY, WSW_SEWER_TYPE, WSW_SEPARATION_FROM_GAS'
        'required', true);
$this->fh->setElementsProperties(
    'WSW_LOCATED_PRIOR, WSW_INSPECTION_NEEDED',
    'input_type', 'y/n');
$this->fh->setElementsProperties(
    'WSW_CITY, WSW_SEWER_TYPE, WSW_SEPARATION_FROM_GAS',
    'input_type', 'select');
$this->fh->setElementsProperties(
    'WSW_NOTES', 'input_type', 'textarea');

etc...

// This creates Zend_Form_Elements out of the meta data
$this->fh->addElementsFromMetaData();
$this->addElements($this->fh->getElements());

// Add a drop-down (<select>) list for Town
$dd = new Code_Values_Master($this->conn);
$ddList = $dd->getCodeValuesList('TOWN', '');
$this->fh->setMultiOptions('WSW_CITY', $ddList);

etc...
<table>
<thead>
<tr>
<th>Values</th>
<th>Drop Down ID</th>
<th>Description</th>
<th>Status</th>
<th>Codes</th>
<th>Last Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPAIR_METHOD_EQUIP</td>
<td>Repair method/equipment for maintenance W/Os</td>
<td>Active</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONDITION_FOUND</td>
<td>Pipe condition found for maintenance W/Os</td>
<td>Active</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONS_TYPES_MI</td>
<td>Construction Types for Main Install</td>
<td>Active</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONS_TYPES_SI</td>
<td>Construction Types for Service Install</td>
<td>Active</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PP_CONTACT</td>
<td>Default contact info for plastic pipe failure</td>
<td>Active</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PP_MANUFACTURER</td>
<td>Plastic Pipe Failure - Manufacturer</td>
<td>Active</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEWER_TYPE</td>
<td>Sewer Types</td>
<td>Active</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREMISE_STS</td>
<td>Premise record status codes (UPRM)</td>
<td>Active</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHOD_OF_CONSTRUCTION</td>
<td>Method of Construction</td>
<td>Active</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WCN_EMAIL_ADDR</td>
<td>Email address to receive W/O cancellation</td>
<td>Active</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WCN_REASON_CODE</td>
<td>W/O Cancellation Reasons</td>
<td>Active</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRI_CUSTOMER_EXCAVATED</td>
<td>Customer Excavated</td>
<td>Active</td>
<td>4</td>
<td></td>
<td>Thu, Jan 12 2012 at 10:14:19 am</td>
</tr>
<tr>
<td>RATECLASS</td>
<td>Rate Classes</td>
<td>Active</td>
<td>6</td>
<td></td>
<td>Mon, Nov 21 2011 at 11:14:32 pm</td>
</tr>
<tr>
<td>WPE_PIPE_COMPOSITION</td>
<td>Pipe Composition</td>
<td>Active</td>
<td>6</td>
<td></td>
<td>Fri, Oct 07 2011 at 04:19:28 pm</td>
</tr>
<tr>
<td>LK_SURVEY_TYPE</td>
<td>Leak Survey Type</td>
<td>Active</td>
<td>5</td>
<td></td>
<td>Fri, Oct 07 2011 at 04:17:14 pm</td>
</tr>
<tr>
<td>AP_PROFILE_TYPE</td>
<td>Authority Profile Type</td>
<td>Active</td>
<td>2</td>
<td></td>
<td>Fri, Aug 12 2011 at 04:12:56 pm</td>
</tr>
<tr>
<td>AP_PERMISSION</td>
<td>Authority/Profile Permissions</td>
<td>Active</td>
<td>4</td>
<td></td>
<td>Fri, Aug 12 2011 at 12:57:28 pm</td>
</tr>
<tr>
<td>AD_Functional_AREA</td>
<td>Grouping ID for Authority Definitions</td>
<td>Active</td>
<td>2</td>
<td></td>
<td>Fri, Aug 12 2011 at 12:50:20 pm</td>
</tr>
<tr>
<td>WO_RETIRED_MAIN</td>
<td>Retired With or At Main</td>
<td>Active</td>
<td>3</td>
<td></td>
<td>Tue, Jul 26 2011 at 11:03:53 am</td>
</tr>
<tr>
<td>LK_EVENTS</td>
<td>Leak Collateral Incidents</td>
<td>Active</td>
<td>3</td>
<td></td>
<td>Mon, Jul 25 2011 at 04:56:37 pm</td>
</tr>
<tr>
<td>WO_METER_LOCATION</td>
<td>Meter Location</td>
<td>Active</td>
<td>4</td>
<td></td>
<td>Mon, Jul 18 2011 at 03:29:02 pm</td>
</tr>
<tr>
<td>PRI_STATUS</td>
<td>Project Status</td>
<td>Active</td>
<td>5</td>
<td></td>
<td>Tue, Jul 12 2011 at 02:29:24 pm</td>
</tr>
<tr>
<td>WO_FLOW_LIMITER_SIZE</td>
<td>Valid Flow Limiter Sizes</td>
<td>Active</td>
<td>3</td>
<td></td>
<td>Fri, Jul 08 2011 at 12:32:24 pm</td>
</tr>
<tr>
<td>MF_MECHANICAL_FITTING</td>
<td>Mech Fitting Failure Fitting</td>
<td>Active</td>
<td>4</td>
<td></td>
<td>Thu, Jul 07 2011 at 02:48:01 pm</td>
</tr>
<tr>
<td>MF_SUPPLEMENTAL_REPORT</td>
<td>Mech Fitting Supplemental Report</td>
<td>Active</td>
<td>2</td>
<td></td>
<td>Wed, Jul 06 2011 at 03:41:47 pm</td>
</tr>
<tr>
<td>MF_INITIAL_REPORT</td>
<td>Mech Fitting Initial Report - Y/N</td>
<td>Active</td>
<td>2</td>
<td></td>
<td>Wed, Jul 06 2011 at 03:39:45 pm</td>
</tr>
</tbody>
</table>
### User Maintainable Drop Down Lists (for `<select>`)

#### VGS Work Order Management System

**Drop Down Values Search**

<table>
<thead>
<tr>
<th>List</th>
<th>Seq#</th>
<th>Code</th>
<th>Value</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEWER_TYPE</td>
<td>1.0000</td>
<td>SEWER</td>
<td>Sewer</td>
<td></td>
<td>Active</td>
</tr>
<tr>
<td>SEWER_TYPE</td>
<td>2.0000</td>
<td>SEPTIC</td>
<td>Septic</td>
<td></td>
<td>Active</td>
</tr>
<tr>
<td>SEWER_TYPE</td>
<td>3.0000</td>
<td>BOTH</td>
<td>Both (sewer/septic)</td>
<td></td>
<td>Active</td>
</tr>
<tr>
<td>SEWER_TYPE</td>
<td>4.0000</td>
<td>NONE</td>
<td>None</td>
<td></td>
<td>Active</td>
</tr>
<tr>
<td>SEWER_TYPE</td>
<td>5.0000</td>
<td>UNKOWN</td>
<td>Unknown</td>
<td></td>
<td>Active</td>
</tr>
</tbody>
</table>
Defines all table specific data access methods

- Parent class (VGS_DB_Table)

Encapsulates basic DB2 functionality:

- Public methods used with Search lists and VGS_Paginator:
  - execListQuery($queryString, $bindParms = array())
  - execScrollableListQuery(VGS_DB_Select $select)
  - getRowCount(VGS_DB_Select $select)

- Public methods used to retrieve and update single record (detail screens)
  - execUpdate($queryString, $bindParms = array())
  - fetchRow($queryString, $bindParms = array())

- Security:
  - checkPermissionByCategory( $category, $mode )
  - Ensure user has authority to table for given mode
VGS_DB_Table - SQL generator methods

autoCreateRecord(array $inputs)
avoUpdateRecord(array $inputs)
avoDeleteRecord(array $inputs)

• Automatically build SQL statements from form inputs
• If form fields change, never need to modify SQL statements
• Never have to align field names and values
• Uses bound parameters - no need to align parameter markers (?)
• Huge time saver; ensures accurate updates without coding
• Bound parameters prevents SQL injection attacks
VGS_DB_Table - public attributes

Name of the database table, specified in UPPERCASE.
public $tableName;

Field names prefix for this table (eg: 'WO_');
Used to extract the update fields from form inputs.
public $tablePrefix;

Array of key field names for this table
public $keyFields;

Boolean = table includes audit fields (Default = true)
public $hasAuditFields;

Boolean = physical record delete is allowed. (Default = false)
public $isRecordDeletionAllowed;

- With above attributes, system can automatically create SQL for create,
  update, delete, and set audit fields appropriately.
Example of VGS_DB_Table based class

class WO_Sewer extends VGS_DB_Table {
    public function __construct($conn) {
        parent::__construct($conn);
        $this->tableName = 'WO_SEWER';
        $this->tablePrefix = 'WSW_';
        $this->keyFields = array('WSW_WO_NUM', 'WSW_SEQNO');
        $this->hasAuditFields = true;
        $this->isRecordDeletionAllowed = true;
    }

    public function create( $rec ) {
        $this->checkPermissionByCategory('WO', 'CREATE');
        $rec['WSW_SEQNO'] = $this->getNextSewerNum($rec['WSW_WO_NUM']);
        $this->autoCreateRecord($rec);
    }

    public function update( $rec ) {
        $this->checkPermissionByCategory('WO', 'UPDATE');
        $this->autoUpdateRecord($rec);
    }
}
Example - Update Sewer Details

### Work Order Details

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/O Number</td>
<td>64901</td>
</tr>
<tr>
<td>W/O Type</td>
<td>SI</td>
</tr>
<tr>
<td>W/O Status</td>
<td>CMP</td>
</tr>
<tr>
<td>Date Completed</td>
<td>Oct 08, 2012</td>
</tr>
</tbody>
</table>

### Method of Construction

<table>
<thead>
<tr>
<th>Method</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOC Trench</td>
<td></td>
</tr>
<tr>
<td>MOC HDD</td>
<td></td>
</tr>
<tr>
<td>MOC Hog</td>
<td></td>
</tr>
<tr>
<td>MOC Plowed</td>
<td></td>
</tr>
<tr>
<td>MOC Other</td>
<td></td>
</tr>
</tbody>
</table>

### Sewer Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>19 Whisper Ln</td>
</tr>
<tr>
<td>City</td>
<td>Milton Village</td>
</tr>
<tr>
<td>Sewer Located Prior</td>
<td></td>
</tr>
<tr>
<td>Sewer Size</td>
<td></td>
</tr>
<tr>
<td>Sewer Material</td>
<td></td>
</tr>
<tr>
<td>Sewer Type</td>
<td>Septic</td>
</tr>
<tr>
<td>Separation From New Gas Install</td>
<td>Rear/Opposite Side</td>
</tr>
<tr>
<td>Inspection Needed</td>
<td></td>
</tr>
</tbody>
</table>

### Comments

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Notes</td>
<td></td>
</tr>
</tbody>
</table>

### Sewer Record Maintenance Info

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created by User ID</td>
<td>KIM</td>
</tr>
<tr>
<td>Date/Time Created</td>
<td>Tue, Oct 05 2012 at 02:06:15 pm</td>
</tr>
<tr>
<td>Last Changed by User ID</td>
<td></td>
</tr>
<tr>
<td>Date/Time Last Changed</td>
<td></td>
</tr>
</tbody>
</table>
Sewer Update - example input values

- Inputs array passed to VGS_Form->autoUpdateRecord( $inputs )
- Red elements are ignored: prefix not = 'WSW_'

```
WSW_WO_NUM = 64901
WSW_SEQNO = 1
popup =
mode = update
a = update
return_point = /wotest/controller/wswListCtrl.php
WO_TYPE = SI
WO_STATUS = CMP
WO_DATE_COMPLETED = Oct 08, 2012
WSW_ADDRESS = 19 Whisper Ln
WSW_CITY = MLV
WSW_LOCATED_PRIOR = N
WSW_SEWER_SIZE =
WSW_SEWER_MATERIAL =
WSW_SEWER_TYPE = SEPTIC
WSW_SEPARATION_FROM_GAS = REAR
WSW_INSPECTION_NEEDED = N
WSW_INSPECT_REASON = in rear
WSW_MOC_TRENCH = Y
WSW_MOC_HDD = Y
WSW_MOC_HOG = N
WSW_MOC_PLOWED = N
WSW_MOC_OTHER =
WSW_NOTES =
```
• Builds SQL update string, and array of values to bind, then...

• $this->execUpdate($sql, $values);

\$sql:

update WO_SEWER set WSW_ADDRESS = ?, WSW_CITY = ?, WSW_LOCATED_PRIOR = ?, WSW_SEWER_SIZE = ?,
WSW_SEWER_MATERIAL = ?, WSW_SEWER_TYPE = ?, WSW_SEPARATION_FROM_GAS = ?, WSW_INSPECTION_NEEDED = ?,
WSW_INSPECT_REASON = ?, WSW_MOC_TRENCH = ?, WSW_MOC_HDD = ?, WSW_MOC_HOG = ?, WSW_MOC_PLOWED = ?,
WSW_MOC_OTHER = ?, WSW_NOTES = ?, WSW_CHANGE_USER = ?, WSW_CHANGE_TIME = current timestamp
where WSW_WO_NUM = ? AND WSW_SEQNO = ?

\$values:

<table>
<thead>
<tr>
<th>WSW_ADDRESS</th>
<th>19 Whisper Ln</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSW_CITY</td>
<td>MLV</td>
</tr>
<tr>
<td>WSW_LOCATED_PRIOR</td>
<td>N</td>
</tr>
<tr>
<td>WSW_SEWER_SIZE</td>
<td></td>
</tr>
<tr>
<td>WSW_SEWER_MATERIAL</td>
<td></td>
</tr>
<tr>
<td>WSW_SEWER_TYPE</td>
<td>SEPTIC</td>
</tr>
<tr>
<td>WSW_SEPARATION_FROM_GAS</td>
<td>REAR</td>
</tr>
<tr>
<td>WSW_INSPECTION_NEEDED</td>
<td>N</td>
</tr>
<tr>
<td>WSW_INSPECT_REASON</td>
<td>in rear</td>
</tr>
</tbody>
</table>

| WSW_MOC_TRENCH   | Y             |
| WSW_MOC_HDD      | Y             |
| WSW_MOC_HOG      | N             |
| WSW_MOC_PLOWED   | N             |
| WSW_MOC_OTHER    |               |
| WSW_NOTES        |               |
| WSW_CHANGE_USER  | JVALANCE      |
| WSW_WO_NUM       | 64901         |
| WSW_SEQNO        | 1             |

• **Red** = audit fields - automatically inserted

• **Green** = key fields, put at end of $values array
JOD Reports

- Java OpenDocument Reports
- http://jodreports.sourceforge.net/
- Open source, Java-based report template tool
- Create documents and reports in OpenDocument Text format from templates
- Templates can be visually composed using the OpenOffice.org Writer word processor
- These documents can then be converted to PDF, Word and RTF with JODConverter
<WO>
  <WO_NUM>65093</WO_NUM>
  <NEED_BY_DATE>Thu Oct 25, 2012</NEED_BY_DATE>
  <WO_DESCRIPTION>47 Barrett St</WO_DESCRIPTION>
  <WO_PREMISE_NUM>27590</WO_PREMISE_NUM>
  <OWNERS_NAME>Valance, John G</OWNERS_NAME>
  <OWNERS_PHONE>802-355-4024</OWNERS_PHONE>
  <METER_NO>25018</METER_NO>
  <WO_SPECIAL_INSTRUCTION />
  <WO_TYPE_DESC>Service New Construction</WO_TYPE_DESC>
  <WO_GL_COST>VGSBS-1071-0-65093</WO_GL_COST>
  <PT_DESCRIPTION>Plastic Service 1"</PT_DESCRIPTION>
  <ESTLEN>.00</ESTLEN>
  <ESTHRS>.00</ESTHRS>
  <CURBSTOP>N</CURBSTOP>
  <FLWLIM>800</FLWLIM>
  <WO_TOWN_NAME>So. Burlington</WO_TOWN_NAME>
  <MAINPIPE_TYPE />
</WO>
**VERMONT GAS SYSTEMS, INC.**

**Work Order for Input field**

<table>
<thead>
<tr>
<th>Date Issued</th>
<th>Input field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Input field - Input field</td>
</tr>
<tr>
<td>Needed By</td>
<td>Input field</td>
</tr>
<tr>
<td>Type of Work</td>
<td>Input field</td>
</tr>
<tr>
<td>Owner's Name</td>
<td>Input field</td>
</tr>
<tr>
<td>Owner's Phone#</td>
<td>Input field</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Charge Acct#</th>
<th>Input field</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROW#</td>
<td>Input field</td>
</tr>
<tr>
<td>Pipe Type</td>
<td>Input field Input field</td>
</tr>
<tr>
<td>Est. Length</td>
<td>Input field</td>
</tr>
<tr>
<td>Est. Crew Hrs</td>
<td>Input field</td>
</tr>
<tr>
<td>Curb Stop</td>
<td>Input field</td>
</tr>
<tr>
<td>Flow Limiter</td>
<td>Input field</td>
</tr>
</tbody>
</table>

**Special Instructions:** Input field

**Dig Safe Auth. No.:**

By:

Begin Dig Date: __________

**Soil:** Sand Clay Rocky Packing: Loose Hard

**Soil Moisture:** Dry Moist Wet N/A

**Exposed MAIN Info**

Exposed MAIN Info Size: 3/4” 1” 1 1/2” 2” 4” 6”

Pipe Type: DPE8000 DPE1000 PPE8300 EndoPE

Coating Type: FlexClad Pritec Xtru Scotch

Coating External Cond. Good Fair Poor Smooth

Internal Cond. Good Fair Poor Smooth Pile

C20 Reading: __________ Main Depth: _______

Clean Up Required: (Dimensions)

- Topsoil
- Blacktop
- Concrete
- Gravel/Stone

Comments: Input field

**Input Field**

```
javascript

$(WO.WO_ENTRY_DATE)
```

**Excavation Permit #**

**Edit**

- OK
- Cancel
- Help

**SlsApp#** Input field

**Premise#** Input field

**Address** Input field

**Apt#** Input field

**Sls Pers.** Input field
public function retrievePDF($xml, $request) {
    // urlenocde and concatenate the POST arguments
    $postargs = 'outputFormat=pdf&model=' . urlencode($xml);
    // $request is JOD-compliant URL for appropriate report template
    $session = curl_init ( $request );
    curl_setopt($session, CURLOPT_POSTFIELDS, $postargs); // this is body of POST
    curl_setopt($session, CURLOPT_RETURNTRANSFER, true); // return response
    return $response;
}
Contact Information:
John Valance
johnv@jvalance.com
802-355-4024