# imagine 2013

Magento® Conference

THE ART OF COMMERCE

# Creating Successful Magento ERP Integrations

CLASSY LLAMA

imagine 2013
Magento Conference

# Happy Together

## Creating Successful Magento ERP Integrations

David Alger

CTO / Lead Engineer

www.classyllama.com

CLASSY LLAMA

imagine 2013

Magento Conference

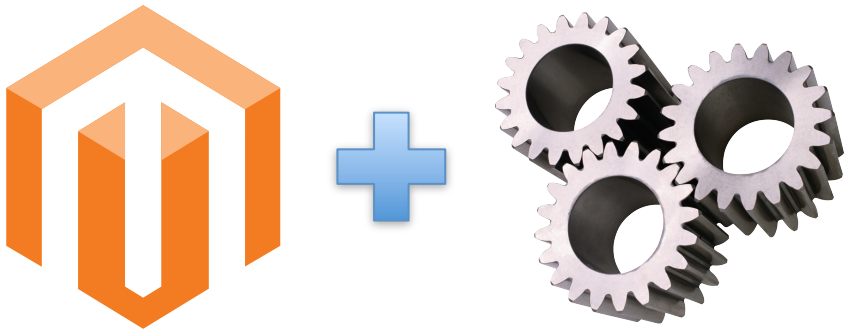
# A Little About Me

- Exclusively focused on Magento eCommerce since early ’09
- Member of the Magento Technical Partner Council
- Member of the Magento Certification Advisory Board
- Magento Certified Developer Plus
- Magento Front End Certified Developer
- Zend Certified Engineer
- Experienced Software Integrator

CLASSY LLAMA

imagine 2013
Magento Conference

# Creating Magento ERP Integrations



Architecting and engineering a Magento ERP integration while avoiding the pitfalls and challenges that come with integrating multiple enterprise-level software components

CLASSY LLAMA

imagine 2013
Magento Conference

# What's On Our Agenda

- Reasons for Integrating
- Purpose Driven Nature
- Architectural Design
- Facing Challenges
- Engineering Solutions
- Integration Scalability

CLASSY LLAMA

imagine 2013
Magento Conference

# Why an Integration?

- An ERP in simple terms:
  - Software built around the premise of having the ability to manage all of the information and functions of a company seamlessly and in real-time
- What benefits are there?
  - Centralization, accuracy, less hassle!
    - Order Fulfillment
    - Customer Service
    - Product Management
- Integrating Magento with your ERP means an increase in efficiency

# Purpose Driven Nature

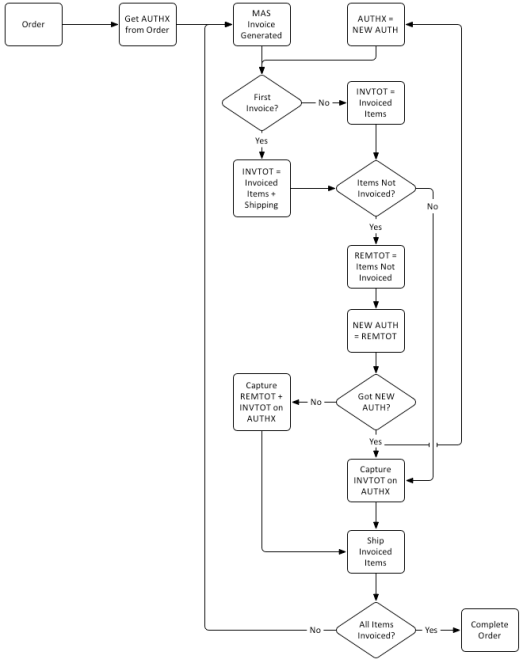
- Integrations with any software should be purpose driven
- What are the goals of the integration?
  - Example: Guaranteed accurate information across properties
  - Example: Less headaches during order fulfillment

# Architectural Design

- The importance of having an integration “blueprint”
  - Information Coverage
  - Master of Record
  - Direction of Flow
- Pre-requisites for designing a workable solution
  - Insights into business process
  - Familiarity with software
  - Knowledge of the data
- Integration paradigms
  - Middleware vs Direct

# Typical Technical “Blueprint”

1. Integration Goals / Summary
2. Integration Dependencies
3. Integration Method
4. Integration Requirements
    - 4.1. Product Information
        - 4.1.1. Inventory
        - 4.1.2. Pricing
            - 4.1.2.1. General Pricing
            - 4.1.2.2. Customer Tiers
    - 4.2. Customer Accounts
    - 4.3. Orders
        - 4.3.1. Export
        - 4.3.2. Fulfillment / Updates
        - 4.3.3. Payment Processing



# Facing Challenges

- Using COTS (Commercial Off the Shelf) products
- Controlling data exposure at connection points
  - aka PCI-DSS compliance
- Real-time data synchronization: Is it possible or even worth it?
- Maintaining data integrity between disparate systems
- Overall reliability of the integration processes

# Engineering Solutions

- Methods of Direct Integration
- Managing Memory Leaks
- Building a Scalable Integration

CLASSY LLAMA

imagine 2013
Magento Conference

# Methods of Direct Integration

- Database
  - Remote
  - Local
- Flat Files
- Web Services
- Combinations

# Managing Memory Leaks

- PHP 5.3 Garbage Collector
  - Largely addressed circular reference memory leaks
  - Far better than PHP 5.2 but doesn't eliminate all possible leaks
  - Global anonymous functions can still be orphaned!
- Looping over all records of a large catalog in one execution is a practical impossibility... particularly when running PHP 5.2 or GC disabled
- Using `memory_get_usage` and a bit of clever logic:
  - We can prevent processes dying due to memory leaks
  - Create a robust and reliable process manager

```
protected static function _getMemLimit() {
    static $limit = NULL;
    if ($limit === NULL) {
        $value = trim(ini_get('memory_limit'));
        $code = strtolower($value[strlen($value)-1]);
        switch ($code) {
            case 'g':   // intentional fall through
                $value *= 1024;
            case 'm':   // intentional fall through
                $value *= 1024;
            case 'k':   // intentional fall through
                $value *= 1024;
        }
        $limit = (int)$value;
    }
    return $limit;
}
```

```php
protected function _processDataCollection($collection, $callback) {
    $index = 0;
    $limit = self::_getMemLimit();        // store the memory limit in bytes for calculations
    $baseline = 0;                        // the current memory usage at last iteration
    $delta = 0;                           // maximum difference in memory usgae from one iteration to the next
    $space = NULL;                        // the remaining number of iterations we have based on the $delta

    foreach ($collection as $record) {
        $baseline = memory_get_usage();    // update the baseline

        try {
            $this->$callback($record);    // process the record
        } catch (Zend_Db_Exception $e) {
            // catch, log and skip items where an exception (like a deadlock or lock timeout) occurs...
            Mage::logException($e);
            continue;
        }

        if ($index == 0) {
            $baseline = memory_get_usage();    // first iteration, update this post-processing to avoid inflated delta
        } else {
            $delta = max($delta, memory_get_usage() - $baseline, 0.0001);    // calculate memory usage delta
            $space = floor(($limit - memory_get_usage()) / $delta);  // calculate approximate space for iterations
        }

        // if we have space for less than 100 estimated iteration remaining, log a message and break to cleanup
        if ($space !== NULL && $space <= 100) {
            Mage::log("CLS_Integration [".__CLASS__."::".__FUNCTION__."]: Must terminate, within 100"
                      ." iterations of remaining space allowed by memory_limit!");
            return false;
        }
    }
    return true;
}
```

# Building a Scalable Integration

- Dealing with large amounts of data
  - Polling for changes
  - Data write throughput
  - Index management
- Managing integration processes
  - Use the built-in cron dispatcher for job execution
  - Thread locks are critical to avoid racing and overworking jobs
  - A page cursor can come in very handy when polling is used

```php
/**
 * Reschedules the cron job $interval seconds into the future.
 *
 * @param Mage_Cron_Model_Schedule $schedule
 * @param int $interval
 */
protected function _rescheduleCron(Mage_Cron_Model_Schedule $pSchedule, $interval = 10) {
    $timestamp = Mage::getSingleton('core/date')->gmtTimestamp()+$interval;
    $schedule = Mage::getModel('cron/schedule');
    $schedule->setJobCode($pSchedule->getJobCode())
        ->setStatus(Mage_Cron_Model_Schedule::STATUS_PENDING)
        ->setCreatedAt($timestamp)
        ->setScheduledAt($timestamp)
    ;
    $schedule->save();

    return $this;
}
```

```
--
--  cls_integration_process_lock - table schema
--

CREATE TABLE `cls_integration_process_lock` (
  `code` varchar(50) NOT NULL COMMENT 'Job Code',
  `locked_at` timestamp NULL DEFAULT NULL COMMENT 'Locked At',
  `freed_at` timestamp NULL DEFAULT NULL COMMENT 'Freed At',
  `status` smallint(6) NOT NULL DEFAULT '0' COMMENT 'Lock Status',
  PRIMARY KEY (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Holds an atomically set lock to prevent overlapping jobs.';
```

```sql
--
--  cls_integration_process_lock - table schema
--

CREATE TABLE `cls_integration_process_lock` (
  `code` varchar(50) NOT NULL COMMENT 'Job Code',
  `locked_at` timestamp NULL DEFAULT NULL COMMENT 'Locked At',
  `freed_at` timestamp NULL DEFAULT NULL COMMENT 'Freed At',
  `status` smallint(6) NOT NULL DEFAULT '0' COMMENT 'Lock Status',
  PRIMARY KEY (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Holds an atomically set lock to prevent overlapping jobs.';


--
--  cls_integration_process_lock - atomic lock update
--  ref: CLS_Integration_Model_Resource_Process_Lock::obtainLock
--

UPDATE `cls_integration_process_lock` SET `status` = 1 WHERE `status` = 0 AND `code` = 'product';
```

P.S. WE'RE HIRING!!!

# Questions or Comments?

David Alger (@blackbooker)
CTO / Lead Engineer

www.classyllama.com
david@classyllama.com

http://bit.ly/imagine-erp

CLASSY LLAMA

imagine 2013
Magento Conference