<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Montage Architecture</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Third Party Components</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Montage Components</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Data Flow</td>
<td>5</td>
</tr>
<tr>
<td>2 Montage Server Requirements</td>
<td>7</td>
</tr>
<tr>
<td>2.1 System Requirements</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Storage Performance</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Supported Operating Systems</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Notes</td>
<td>8</td>
</tr>
<tr>
<td>3 Installation Checklists</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Pre-Install Checklist</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Post-Install Checklist</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Nuance Implementation Checklists</td>
<td>11</td>
</tr>
<tr>
<td>3.4 PowerScribe 360 Data Analysis Script</td>
<td>13</td>
</tr>
<tr>
<td>4 Windows Installation Guide</td>
<td>17</td>
</tr>
<tr>
<td>4.1 Dependency Installation</td>
<td>17</td>
</tr>
<tr>
<td>4.2 Initial Configuration</td>
<td>47</td>
</tr>
<tr>
<td>4.3 Montage Installation</td>
<td>74</td>
</tr>
<tr>
<td>4.4 Post Installation Configuration</td>
<td>80</td>
</tr>
<tr>
<td>4.5 Initial Crawl</td>
<td>88</td>
</tr>
<tr>
<td>4.6 Windows Frequently Asked Questions</td>
<td>91</td>
</tr>
<tr>
<td>4.7 Troubleshooting</td>
<td>92</td>
</tr>
<tr>
<td>5 Source System Crawlers</td>
<td>95</td>
</tr>
<tr>
<td>5.1 General Crawler Configuration</td>
<td>95</td>
</tr>
<tr>
<td>5.2 Database URL</td>
<td>99</td>
</tr>
<tr>
<td>5.3 Nuance</td>
<td>99</td>
</tr>
<tr>
<td>5.4 Montage HL7 Interface Specification</td>
<td>102</td>
</tr>
<tr>
<td>5.5 Montage File Format Extract Specification</td>
<td>104</td>
</tr>
<tr>
<td>6 Settings</td>
<td>109</td>
</tr>
<tr>
<td>6.1 Authentication</td>
<td>109</td>
</tr>
<tr>
<td>6.2 Crawlers</td>
<td>110</td>
</tr>
<tr>
<td>6.3 Email</td>
<td>110</td>
</tr>
<tr>
<td>6.4 External Launchers</td>
<td>111</td>
</tr>
<tr>
<td>6.5 File Locations</td>
<td>113</td>
</tr>
<tr>
<td>6.6 General Settings</td>
<td>114</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.7 Group Organizations</td>
<td>115</td>
</tr>
<tr>
<td>6.8 Indexes</td>
<td>116</td>
</tr>
<tr>
<td>6.9 LDAP / ActiveDirectory</td>
<td>116</td>
</tr>
<tr>
<td>6.10 Analytics</td>
<td>119</td>
</tr>
<tr>
<td>6.11 Purpose of Use</td>
<td>120</td>
</tr>
<tr>
<td>6.12 Radimetrics eXposure</td>
<td>120</td>
</tr>
<tr>
<td>7 montage Command Line</td>
<td>123</td>
</tr>
<tr>
<td>7.1 Commands</td>
<td>123</td>
</tr>
<tr>
<td>8 Backing up Montage</td>
<td>125</td>
</tr>
<tr>
<td>8.1 Windows PostgreSQL Backup</td>
<td>125</td>
</tr>
<tr>
<td>8.2 Linux PostgreSQL Backup</td>
<td>126</td>
</tr>
<tr>
<td>9 Additional Montage Tools</td>
<td>127</td>
</tr>
<tr>
<td>9.1 Truncate (Wipe) the Database</td>
<td>127</td>
</tr>
<tr>
<td>10 Montage Privacy &amp; Security</td>
<td>129</td>
</tr>
<tr>
<td>10.1 System Architecture</td>
<td>129</td>
</tr>
<tr>
<td>10.2 Security Model</td>
<td>129</td>
</tr>
<tr>
<td>10.3 Data Model</td>
<td>130</td>
</tr>
<tr>
<td>10.4 Frequently Asked Questions</td>
<td>130</td>
</tr>
<tr>
<td>11 Firewall</td>
<td>131</td>
</tr>
<tr>
<td>11.1 ufw on Ubuntu</td>
<td>131</td>
</tr>
<tr>
<td>11.2 Windows Firewall</td>
<td>131</td>
</tr>
<tr>
<td>12 Montage Log Files</td>
<td>133</td>
</tr>
<tr>
<td>12.1 montage-audit.log</td>
<td>133</td>
</tr>
<tr>
<td>12.2 montage.log</td>
<td>133</td>
</tr>
<tr>
<td>12.3 montage-importer.log</td>
<td>133</td>
</tr>
<tr>
<td>12.4 montage-importer-hl7.log</td>
<td>133</td>
</tr>
<tr>
<td>12.5 searchd.log</td>
<td>133</td>
</tr>
<tr>
<td>12.6 Apache Logs</td>
<td>133</td>
</tr>
<tr>
<td>13 Frequently Asked Questions</td>
<td>135</td>
</tr>
<tr>
<td>13.1 Can Montage be secured with HTTPS / SSL?</td>
<td>135</td>
</tr>
<tr>
<td>13.2 Can Montage be scaled or clustered?</td>
<td>135</td>
</tr>
<tr>
<td>13.3 Can Montage crawl our RIS / LIS / HIS / EMR / etc.?</td>
<td>136</td>
</tr>
<tr>
<td>13.4 Can Montage use a different database or webservice?</td>
<td>136</td>
</tr>
<tr>
<td>13.5 What authentication systems does Montage support?</td>
<td>136</td>
</tr>
<tr>
<td>13.6 How often does Montage crawl / index data?</td>
<td>137</td>
</tr>
<tr>
<td>13.7 Does Montage perform Backups?</td>
<td>137</td>
</tr>
<tr>
<td>14 Support</td>
<td>139</td>
</tr>
<tr>
<td>15 Montage Glossary</td>
<td>141</td>
</tr>
<tr>
<td>16 Changelog</td>
<td>143</td>
</tr>
<tr>
<td>16.1 3.0 Changelog</td>
<td>143</td>
</tr>
<tr>
<td>16.2 2.5 Changelog</td>
<td>144</td>
</tr>
<tr>
<td>16.3 2.4 Changelog</td>
<td>147</td>
</tr>
<tr>
<td>16.4 2.3 Changelog</td>
<td>149</td>
</tr>
<tr>
<td>16.5 2.2 Changelog</td>
<td>152</td>
</tr>
<tr>
<td>16.6 2.1 Changelog</td>
<td>152</td>
</tr>
<tr>
<td>16.7 2.0 Changelog</td>
<td>153</td>
</tr>
</tbody>
</table>
The Montage Server Manual is intended only for people trained to install and manage Montage Servers. This document should not be shared with individual customers, since APIs may change and the may not receive notification of the changes. The Montage Manual, which is distributed as part of the Montage Installation contains a section specifically designed for customers on administering the Montage server.

**Warning:** This document contains intellectual property belonging to Montage Healthcare Solutions, Inc. Distribution of this document must be authorized by Montage Healthcare Solutions, Inc. It is intended only for Montage Support and Montage’s partners and may not be shared with customers without Montage’s authorization.
The Montage application is comprised of several components, each with a specific purpose, allowing Montage crawl multiple vendors’ systems and deliver search and analytics results in milliseconds.

1.1 Third Party Components

1.1.1 Database - PostgreSQL

PostgreSQL is a fast and powerful open-source database server. Montage uses PostgreSQL as its database server to store user settings, crawled reports, terminologies, and various other data elements.
1.1.2 Web Server - Apache

Apache is an open-source web server. When a user’s browser requests a web page, Apache handles the request. If the request is for a static resource, such as a javascript or CSS file or an image, Apache will directly serve the file. If the request is for one of Montage’s dynamic pages, Apache, using the WSGI protocol, hands the request off to one of the running Montage Server processes.

1.1.3 Task Queue - RabbitMQ

RabbitMQ is an open-source messaging server. Montage uses RabbitMQ to process offline jobs, such as crawling upstream report systems, analyzing reports, and cleaning up stale data.

1.1.4 Cache Server - Memcache

Memcache is a high performance, in-memory, cache server. Montage will often perform expensive computations that are frequently requested. Instead of re-computing the computations on every request, Montage caches the result of the computation after the first request and delivers the cached result to subsequent requests.

Typically, many results are cached for at least 2 hours, though this value can be tuned in the Settings.

Note: On Windows, Montage uses a file-based cache instead of Memcache, due to Memcache’s dependency on a high-performance feature found only on Linux. The cache files are typically stored in C:\montage\data\cache. The file-based cache, while not as performant as Memcache, still dramatically improves the overall speed of Montage as opposed to using no cache at all.

1.2 Montage Components

1.2.1 Crawler

The Montage Crawler provides a framework for interfacing with existing source systems, so that Montage can index the report text inside those source systems. Montage provides adapters for each type of source system. The adapter can be a database connection (via ODBC or similar protocol), API, HL7 stream, or a flat file (XML, CSV, plain text). The Crawler framework has proven extremely flexible to date.

The crawler is designed to be efficient and only process reports in the source system that have been changed or added since the last crawling event.

1.2.2 Celeryd

The Montage Celeryd processes provide workers for tasks put on the RabbitMQ Task Queue. These workers allow data to be processed asynchronously (when the server has free CPU cycles).

1.2.3 Celerybeat

Montage Celerybeat is a specialized version of Celeryd. Celerybeat kicks off specific tasks at configured intervals. One such task is the scheduled crawling of source systems.

Celerybeat is conceptually similar to cron of Window’s Scheduled Tasks, but allows finer control of tasks.
1.2.4 Indexer

The Indexer is a program that takes the results of the Crawler and generates a set of highly optimized index files. Currently, the indexer rebuilds the entire set of index files once a day. However, it is possible to perform delta-indexing on a much shorter schedule.

1.2.5 Searchd

The Searchd process is a service that loads the index files into memory, efficiently answering queries.

1.2.6 Website / REST API

The Website & REST API handle HTTP requests from the Apache webserver, by performing queries against Searchd and by retrieving data from the cache server and PostgreSQL. The Website is the primary interface for users of Montage. The REST API provides a programmable method for other applications to internally display the Montage results.

1.3 Data Flow

For demonstration purposes, we will describe the flow of a single radiology report (a Chest CT) through the components that comprise the Montage Server. In our example, the report lives in a Nuance PowerScribe 360 database.

1. The Montage Crawler connects to PowerScribe 360’s SQL Server database via ODBC. It finds one new report, our Chest CT. The Crawler normalizes the report, then places it into the PostgreSQL database and also puts it onto the RabbitMQ task queue

2. The copy of the Chest CT that is in the Task Queue will be picked up by one of the Celeryd processes, which will perform further offline analysis of the report. If Celeryd finds anything interesting, it will save that fact to the PostgreSQL database.

3. The Indexer will run according to its schedule. It will find the Chest CT in the PostgreSQL database. The indexer will perform its natural language processing on the report, saving the results into the Index Files. When the Indexer is finished, it notifies the Searchd service that the Index Files have been updated

4. Searchd will load the updated Index Files into memory, preparing itself to be queried

5. Now a user opens her web browser to the Montage Website. After logging in, she performs a search for “pneumonia”.

6. The Montage website validates the query and passes it onto Searchd. Searchd looks for all reports containing “pneumonia”. It finds our Chest CT report, and responds to the Montage Website with the match.

7. The Montage website wraps the matched report inside an HTML webpage and delivers the result to the user. The Montage website will also cache much of the data to make subsequent results much quicker.
MONTAGE SERVER REQUIREMENTS

2.1 System Requirements

<table>
<thead>
<tr>
<th></th>
<th>Up to 150,000 reports/year</th>
<th>Up To 1,000,000 reports/year</th>
<th>Up To 5,000,000 reports/year</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>64-bit, two-cores, 2.4 GHz minimum per core</td>
<td>64-bit, four-cores, 2.5 GHz minimum per core</td>
<td>64-bit, four-cores, 2.5 GHz minimum per core</td>
</tr>
<tr>
<td>Memory</td>
<td>8GB</td>
<td>16GB</td>
<td>24GB</td>
</tr>
<tr>
<td>Storage Capacity</td>
<td>50GB</td>
<td>200GB</td>
<td>400GB</td>
</tr>
<tr>
<td>Network</td>
<td>100 MBps</td>
<td>1000 Mbps</td>
<td>1000 Mbps</td>
</tr>
</tbody>
</table>

2.2 Storage Performance

Montage is a data-intensive application that requires the ability to quickly scan and analyze millions of records to deliver answers to the questions its users ask. The biggest factor in end-user performance of Montage is the underlying performance of the storage layer.

Montage recommends Tier-0 storage.

The exact configuration (e.g. SAN vs Local RAID) is up to the customer based upon their storage infrastructure. But as an example, we recommend 36,000 IOPS, assuming 80% Read / 20% Write, which would be satisfied by a 4-disk RAID10 SSD array, with total array throughput at roughly 281 MB/s.

2.3 Supported Operating Systems

- Ubuntu 14.04 LTS
- Windows Server 2012 R2

---

1. Assuming 30 years of prior & future data. Can allocate less if storage can be dynamically added later existing partitions (e.g. additional physical volumes to the same logical volume in LVM). Require 10KB (kilobytes) per report, with a minimum of 50GB usable capacity.
2. Total useable capacity after any RAID policies are applied, this does not include the Operating System installation use.
3. Any new installs of Montage v3.0 or greater, as of October 1st, 2015, will only be supported on the listed supported operating systems. Prior installations on Windows 2008 R2 and Ubuntu 10.04 are only supported for customers who went live prior to October 1st, 2015, with Montage <=v2.5. Only these existing live customers are eligible for upgrades to Montage v3.0 on Windows 2008 R2 or Ubuntu 10.04. As of January 1st, 2017, Windows 2008 R2 and Ubuntu 10.04 will not be supported for any version of the Montage application.
• RHEL 6.0 (or comparable CentOS)

2.4 Notes

• Environments may contain bare metal servers or virtualized environments. The above listed should be considered as minimum to operate the Montage system within either type of environment.
CHAPTER THREE

INSTALLATION CHECKLISTS

3.1 Pre-Install Checklist

- System Requirements
- Database Crawler
- HL7 Listener
- Montage Configuration
- Optional Components
  - CPT Searching / RVU Analytics

If applicable, also reference *Nuance Implementation Checklists*.

3.1.1 System Requirements

- Server hardware meets the *Montage Server Requirements*
- Network configuration requirements:
  - The server must have a static IP assigned to it and configured
  - The server must have a domain name, preferably using the *montage* sub-domain, e.g. montage.example.com. Most likely, this should only be visible inside the customer’s network (via an A record or CNAME).
  - The server must be able to connect outbound to https://stats.montagehealthcare.com
  - The server should be able to receive inbound traffic to at least ports 80 (http) and 443 (https). Also if HL7 is enabled, port 6661 (MLLP) should be open for inbound traffic. Other ports should be services which assume they are running in a safe-environment. RDP, SSH, and other required services are fine to open if needed.

3.1.2 Database Crawler

- What is the source system Montage will crawl?
- Vendor & Version?
- Does Montage already have a crawler for this system? If not, Montage Sales & Support must be consulted prior to a sale to investigate and price the crawler and historical migration.
- Need to collect the following database information:
- Vendor & Version (e.g. MS SQL Server 2008R2)
• Hostname (or IP) of the database server
• Database name (e.g. Comm4)
• Database account (user & password), with privileges to perform read-only queries (i.e. SELECT).

If applicable also reference the PowerScribe 360 Crawler checklist.

3.1.3 HL7 Listener

Note: HL7 interfaces must be approved by Montage Sales & Support prior to a sale, since extra professional services are involved.

• What is the sending system?
• Does the site utilize an interface engine, in which they can transform the message?
• Need de-identified sample HL7 messages representing any sending systems and formats.

3.1.4 Montage Configuration

• Has a license been requested from Montage Support? Including montage-modules.json and basic settings.py for this customer.
• If the customer requires authentication other than Montage’s internal password system, such as LDAP or ActiveDirectory, has the proper Information Required from Customer been collected?
• Who should have user accounts in Montage? Need to have the username, first name, last name, and email address. Note for PowerScribe 360, there are additional requirements that the usernames must be the same in PS360 and Montage.
• Which user accounts should have access to the Montage Admin to set up additional users?

3.1.5 Optional Components

CPT Searching / RVU Analytics

The site must provide a lookup table (aka chargemaster) linking Exam Codes and Descriptions to CPTs. Montage Support can then create a conversion script to upload the chargemaster into Montage. This file should be a CSV or XLS file, with the Exam Code and Exam Description in the first 2 columns. The CPT should be in the third column. If multiple CPTs map to an individual Exam Code, each additional CPT should be in a separate column.

Montage Support can then assist in loading and configuring the settings.py to display the CPTs and RVUs.

3.2 Post-Install Checklist

• User accounts
• Have user accounts been setup? Are users able to log in?
• Does at least one person at the customer site have an admin account to set up additional users?
• Crawler / HL7
• Is the crawler / HL7 set up and working
• After 24-hours, has the crawler successfully run and crawled new reports?
• Is the Montage server accessible to workstations at the customer location?
If applicable, also go through the Nuance post-install checklist.

3.2.1 Checkout Procedure

Log into Montage

Montage Search checkout:
1. Conduct a search (e.g. “pneumonia”). Results should be returned.
2. Date filtering should work and provide data (data should be current for the Last 7 days)
3. Are Modalities display (top bar or in the Advanced Search), do these match what the customer expects?

Montage Analytics checkout:
1. Go to Montage Analytics > Practice Overview. Volume & TAT should be displayed. If CPTs have been configured, RVUs should display.
2. Build a Custom Graph, select “Date Interval” as the Concept and “Volume” as the Measurement. View the graph, should have data from past 30 days. Adjust filters to see graph update

Montage QC checkout. Requires QC to be enabled on the server. Should wait until the nightly has run:
1. Go to Montage QC, are there critical and mismatch events?

3.2.2 Optional

• Has the site requested backups, see Backing up Montage

3.3 Nuance Implementation Checklists

Please references Pre-Install Checklist and Post-Install Checklist for the general checklists. This document adds several items to check during Nuance PowerScribe 360 implementations.

• Additional PowerScribe 360 Pre-install Checklist
  – PowerScribe Crawler Checklist
    * Site vs SiteLocation
    * PatientClass
  – Suboptimal Modality Mapping
• Additional PowerScribe 360 extra Post-install Checklist
  – Launching Montage from PowerScribe 360

3.3.1 Additional PowerScribe 360 Pre-install Checklist

PowerScribe Crawler Checklist
Montage ships with sane defaults for the PowerScribe 360 crawler, but there are several common alterations that sites have requested.
Site vs SiteLocation

Montage by default pulls the PowerScribe 360 Site table into the Organization concept. Ideally in PowerScribe, if there are multiple facilities, they would each be set up as a separate Site in PowerScribe 360.

However, Montage does offer the ability to use PowerScribe 360's SiteLocation instead of the Site. Note, that at some customer sites SiteLocation can be overly specific. Therefore, these queries have been provided to assist in comparing Site and SiteLocation prior to crawling.

Montage Support can assist but the feature is enabled via use_site_location_as_organization.

PatientClass

Montage currently supports 3 patient statuses, Inpatient (I), Outpatient (O), and Emergency (E). However, PowerScribe supports several additional PatientClass values (from the HL7 specification), including Pre-admit (P), Recurring Patient (R), Obstetrics (B), Commercial Account (C), Not Applicable (N), Unknown (U). If these other values are used in ExplorerDistinct.PatientClassID (joined to PatientClass table), the customer can provide a mapping from these values into I/E/O.

Note: PowerScribe 360 itself has a strict definition of acceptable PatientClass values and often the RIS will send an out-of-spec value, which PS360 will store as NULL in ExplorerDistinct.PatientClassID. If this is the case, a modification must also be made in the PS360 interface to map the non-spec values into valid PS360 PatientClass values.

Before attempting to customize the patient status mapping, it may be useful to list the existing patient statuses. If needed, Montage Support can assist with mapping, which goes into patient_status.

Suboptimal Modality Mapping

Some PowerScribe 360 have too many modalities (duplicates / legacy values).

Montage can disable crawling the modality from PowerScribe 360, but requires the site to provide a mapping between Exam Codes and Descriptions to Modality.

This can be done with import_modality crawler setting.

3.3.2 Additional PowerScribe 360 extra Post-install Checklist

Launching Montage from PowerScribe 360

Note: Usernames must match between PowerScribe 360 and Montage for the auto-login feature to work.

In RadPortal, 3 settings must be configured to enable the Montage integration with PS360:

Montage Secret Key A secret key generated by Montage Support that allows PS360 to securely authenticate a user within Montage. Note, in settings.py, this is stored at NUANCE_AUTH_SECRET_KEY. The Secret Key should not have any quotes, nor should it have any leading or trailing spaces.

Montage Patient URL The URL template for showing the Patient Timeline. Change montage.example.com as needed, but leave the {mrn} / {psuser} / etc as PS360 will use them to substitute the appropriate context at dictation time:
Montage Search URL. The URL template for conducting queries. Change montage.example.com as needed:

```
http://montage.example.com/search/rad?q={question}&psuser={psuser}&pstime={pstime}&pstoken={pstoken}
```

## 3.4 PowerScribe 360 Data Analysis Script

The following queries have been provided to simplify the decision making process prior to performing a Montage installation at a PowerScribe 360 site. These queries will help identify common situations that may require a Montage configuration other than default. Involving Montage support or the customer will often still be necessary to determine the best course of action if a potential problem has been identified.

SQL queries in this section can be executed in Microsoft SQL Server Management Studio. To execute a single query, simply copy and paste the code from this document into Microsoft SQL Server Management Studio. Next click "Execute".

### 3.4.1 Listing Sites and SiteLocations

In some cases SiteLocation may be preferred over Site as described in Site Vs SiteLocation. The below query will assist in providing a list from both the Site and SiteLocation to simplify the decision making process.

```
-- Site Vs SiteLocation
SELECT Name AS "Site" FROM Site WITH (NOLOCK);
SELECT Name AS "Site Locations" FROM SiteLocation WITH (NOLOCK);
```

<table>
<thead>
<tr>
<th>Site</th>
<th>Site Location 1</th>
</tr>
</thead>
</table>

Table 3.1: SQL Results
3.4.2 Patient Status

Montage will need to be configured to map unrecognized patient statuses if any exist. Please reference Patient Class for more information. The following query will assist in this decision making processes by listing all the patient statuses that have at least one reference from a report followed by the number of times that the status appears in all reports.

```
-- Patient status name and count
SELECT PatientClass.Name AS "Patient Status",
       COUNT(PatientClass.Name) AS "Count"
FROM PatientClass
     JOIN ExplorerDistinct ON PatientClass.PatientClassID = ExplorerDistinct.PatientClassID
GROUP BY ExplorerDistinct.PatientClassID, PatientClass.Name
```

Table 3.3: SQL Results

<table>
<thead>
<tr>
<th>Patient Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinic</td>
<td>2000</td>
</tr>
<tr>
<td>Discharged</td>
<td>45</td>
</tr>
<tr>
<td>Emergency</td>
<td>999</td>
</tr>
<tr>
<td>Inpatient</td>
<td>4001</td>
</tr>
</tbody>
</table>

In the above SQL Results, we find that there two unrecognized patient statuses that need to have a mapping defined for them.

3.4.3 Modalities

The following queries can assist when making decisions about available modalities and other related configurations.

**List Modalities**

The following query can be used to identify modalities that are less common and may need to be clarified by the customer.

```
-- Available modalities
SELECT COUNT(Name) AS "Order Count",
              Name AS "Modality Name"
FROM ProcedureModality
     WITH (NOLOCK)
     JOIN OrderProcedure
          WITH (NOLOCK)
GROUP BY ProcedureModality.Name
ORDER BY ProcedureModality.Name
```
The above query will return a list of the modalities which are recognized by this customer’s PS360 installation. At the customer’s discretion, it is possible to map unwanted, legacy, or duplicate modalities to another more common modality such as CR, CT, MR, US, XA, DF, NM, PT, and MG.

In the below SQL results, modalities CT and US are commonly recognized modalities and are according to Order Count are used more frequently at this customer site. However, US-procedure and US-duplex can be considered duplicates of the more common US. Since US appears far more frequently, it might make sense to consolidate the other modalities into UT.

Table 3.4: SQL Results

<table>
<thead>
<tr>
<th>Order Count</th>
<th>Modality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>CT</td>
</tr>
<tr>
<td>1050</td>
<td>US</td>
</tr>
<tr>
<td>39</td>
<td>US - Duplex</td>
</tr>
<tr>
<td>60</td>
<td>US - Procedure</td>
</tr>
</tbody>
</table>

In this next example, MR and MRI both exist as modalities, but its likely that they can merged into a single modality. Since MR appears more frequently, it might be safe to assume MRI can be mapped to MR. The same situation exists for CTA and Angiography, which in many cases could simply be reduced to CT.

Table 3.5: SQL Results

<table>
<thead>
<tr>
<th>Order Count</th>
<th>Modality</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Angiography</td>
</tr>
<tr>
<td>3000</td>
<td>CT</td>
</tr>
<tr>
<td>500</td>
<td>CTA</td>
</tr>
<tr>
<td>2040</td>
<td>MR</td>
</tr>
<tr>
<td>1000</td>
<td>MRI</td>
</tr>
</tbody>
</table>

For assistance with consolidating modalities please contact Montage Support.

**Modality to Exam Mappings**

For assistance with determining if the existing mappings from “Exam Code” to “Modality” are correct the following query can be used.

```sql
SELECT ProcedureModality.Name AS "Modality",
       Explorer.Procedures AS "Exam Code",
       ProcedureDesc AS "Exam Description"
FROM Explorer
JOIN OrderProcedure ON OrderProcedure.OrderID = Explorer.OrderID
WHERE Procedures IS NOT NULL AND ProcedureDesc IS NOT NULL
GROUP BY ProcedureModality.Name, Explorer.Procedures, Explorer.ProcedureDesc
```

**3.4.4 Duplicate Providers**

For both ordering and reporting providers, we want to identify records that are potential duplicates. Since its plausible that two or more providers have the same name, its still necessary to work with the customer to verify the possible redundancy. If a determination is made that the provider has a duplicate entry, it may be necessary to work with Montage Support to clean the data.

The following query will list all ordering and reporting providers.
-- All ordering and reporting providers
SELECT PersonalInfo.LastName AS "Last Name",
       PersonalInfo.FirstName AS "First Name"
FROM "Order" WITH (NOLOCK)
JOIN Physician WITH (NOLOCK) ON Physician.PhysicianID = "Order".ProviderPhysID
JOIN PersonalInfo WITH (NOLOCK) ON PersonalInfo.PersonalInfoID = Physician.PersonalInfoID
WHERE "Order".ProviderPhysID IS NOT NULL
GROUP BY PersonalInfo.LastName, PersonalInfo.FirstName
ORDER BY PersonalInfo.LastName

SELECT SignerLastName AS "Last Name",
       SignerFirstName AS "First Name"
FROM ExplorerDistinct WITH (NOLOCK)
WHERE SignerLastName IS NOT NULL
GROUP BY SignerLastName, SignerFirstName
ORDER BY SignerLastName

Duplicate Ordering Providers

The following query will list the number of possible duplications for each ordering provider.

-- Possible duplicate ordering providers
SELECT COUNT(DISTINCT "Order".ProviderPhysID) - 1 AS "Possible Duplicates",
       PersonalInfo.LastName AS "Last Name",
       PersonalInfo.FirstName AS "First Name"
FROM "Order" WITH (NOLOCK)
JOIN Physician WITH (NOLOCK) ON Physician.PhysicianID = "Order".ProviderPhysID
JOIN PersonalInfo WITH (NOLOCK) ON PersonalInfo.PersonalInfoID = Physician.PersonalInfoID
WHERE "Order".ProviderPhysID IS NOT NULL
GROUP BY PersonalInfo.LastName, PersonalInfo.FirstName
HAVING COUNT(DISTINCT "Order".ProviderPhysID) > 1
ORDER BY PersonalInfo.LastName

<table>
<thead>
<tr>
<th>Possible Duplicates</th>
<th>Last Name</th>
<th>First Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Smith</td>
<td>John</td>
</tr>
<tr>
<td>1</td>
<td>Tudor</td>
<td>Henry</td>
</tr>
<tr>
<td>4</td>
<td>White</td>
<td>Jessica</td>
</tr>
</tbody>
</table>

Table 3.6: SQL Results

The above results tell us that there are 2 ordering providers with a duplicate record and another with 4 duplicate records.

Duplicate Reporting Providers

The following query will list the number of possible duplications for each reporting provider referenced from in a report.

-- Possible duplicate reporting providers
SELECT COUNT(DISTINCT SignerAcctID) - 1 AS "Possible Duplicates",
       SignerLastName AS "Last Name",
       SignerFirstName AS "First Name"
FROM ExplorerDistinct WITH (NOLOCK)
GROUP BY SignerLastName, SignerFirstName
HAVING COUNT(DISTINCT SignerAcctID) > 1
ORDER BY SignerLastName
montage-install-package.zip contains all the tools required to install and manage a Montage Server installation on Windows.

The current Montage installer aims to standardize and simplify the installation process. However, the installer, at this point, is not a fully “one-click” process—it does require the administrator to install a number of dependencies and edit several configuration files. Future versions of the installer will aim to further simplify this process.

### 4.1 Dependency Installation

- Python
- PostgreSQL
- RabbitMQ
- Apache HTTPD
- Next Steps

Unlike on Linux platforms, which have built-in package management capabilities, we must manually install Montage’s dependencies on Windows.

#### 4.1.1 Python

Montage is mostly written in the Python programming language. Similar to .NET applications, we must install the Python 2.7 runtime.

Run the installer, `python-2.7.2.msi`, accepting the defaults. “Install for all users”, click “Next”: 
Install into C:\Python27, click “Next”

**Warning:** Due to how Montage is currently built, Python must be installed at C:\Python27.
Montage uses the default selection of Python features, click “Next”: 
Python will now be installed:
When the installation is finished, exit the installer via “Finish”: 
Note: Prior to Montage 2.4, several Python libraries had to be installed. As of Montage 2.4, these are now included as part of the montage-VERSION.exe installer.

If previously installed, you can safely remove these libraries under the Control Panel > Programs > Uninstall Program. Only remove:

- “Python 2.7 lxml-2.3.1”
- “Python 2.7 psycopg2-2.4”
- “Python 2.7 pyodbc-2.1.11”
- “Python 2.7 python-ldap-2.3.13”
- “Python 2.7 pywin32-217”

Do not remove “Python 2.7.x”.

4.1.2 PostgreSQL

Run the PostgreSQL installer, postgresql-9.1.9-1-windows.exe. Click “Next” to start:
You can install PostgreSQL anywhere, the default is typically fine:
You can additionally place the PostgreSQL data files (the primary consumer of space), anywhere, such as on another drive. However, the default is acceptable:
Pick a password for the Windows postgres user (make a note of this password):
Use the default port 5432 (otherwise, Montage must be configured to point at a non-default port):
Set the locale to *English, United States*:
We are now ready to install, click “Next”: 
PostgreSQL will take several minutes to fully install:
Uncheck the Stack Builder, it does not add anything Montage requires. Click “Finish” to exit the installer.
Later we will create the database, but for now, continue on to install RabbitMQ.

### 4.1.3 RabbitMQ

To install RabbitMQ, we must first install Erlang, the runtime that RabbitMQ runs on, by running `otp_win32_R14B04.exe`. The default component selection is good, click “Next”: 
The default location is fine:
The location of the icons is not important, so we can just click “Next”: 
The installer will run for several minutes:
When the installation is completed, click “Finish”: 
Finally, we run the RabbitMQ installer, `rabbitmq-server-2.6.1.exe`. The default component selection (including the “RabbitMQ Service”) is good, we click “Next” to start the install:
The default location is fine:
The installer should take less than a minute, when completed, click “Finish”: 
RabbitMQ runs as a Windows Service and requires no additional configuration.

### 4.1.4 Apache HTTPD

**Warning:** Apache HTTPD is an alternative webserver to Microsoft IIS. IIS **must not** be installed on the Montage server, or else it will cause Apache to be unable to bind to port 80 to host the Montage application.

Run the *Apache* installer, httpd-2.2.21-win32-x86-openssl-0.9.8r.msi. Click “Next”
Accept the license and click “Next”: 
We are presented with a description of Apache, click “Next”: 

4.1. Dependency Installation
As part of the Pre-Install Checklist, will have a domain name that the server will run as (e.g. montage.example.com). Enter the top-level domain (e.g. example.com) for the Network Domain and the Montage domain (e.g. montage.example.com) for the Server Name. You may also enter an email address. This address will only ever be displayed if the server is improperly configured and Montage can not load. Install Apache “for All Users, on Port 80”. Click “Next”:
Select the “Typical” installation (we do not need the Apache development header files), click “Next”: 

4.1. Dependency Installation
The default location is required. Click “Next”:

**Warning:** The Montage installer will automatically register itself with Apache, but requires that Apache is installed in this default location on C:\
Click “Install” to start the installation:
The install should take less than a minute, when done, click “Finish”
Apache will run as a Windows Service.

4.1.5 Next Steps

At this point we are done installing the dependencies, we can now proceed onto Initial Configuration, before we run the Montage installer.

4.2 Initial Configuration

The Montage installer expects a certain level of configuration to exist before it is run, so that it can use these settings to set up the Montage installation

- Create the PostgreSQL User & Database
- Montage Settings & License
  - montage-modules.json - License
  - settings.py - Main Config File
- Windows Firewall
- Next Steps
4.2.1 Create the PostgreSQL User & Database

The first item of configuration is to create a user and database in PostgreSQL for Montage to connect with and use.

We could use the command-line `psql` tool, but we will demonstrate using the GUI-based `pgAdmin III` tool, which was installed when we installed PostgreSQL.

To open `pgAdmin`, **Start → All Programs → PostgreSQL 9.1 → pgAdmin III**

`pgAdmin` will display all servers it knows about (by default it should display “PostgreSQL 9.1 (x86) (localhost:5432)”, which we recently installed. The red X indicates that `pgAdmin` is not currently connected to that server, so let’s connect by double clicking on that server.

We will be prompted to enter the password for the `postgres` user that we *previously configured*: 
We should now be connected to the PostgreSQL server. If we expand the “Databases” and “Login Roles” trees in the left pane, we will see that only the default “postgres” user and database exist:
Let’s first create a new user, “montage”, by right clicking on “Login Roles” and then clicking on “New Login Role...”:
On the default tab, “Properties”, we enter the “Role name” as “montage”:
Switch to the “Definition” tab to set the “montage” user’s password. Since the firewall will block any external requests to the PostgreSQL server, the password does not have to be extremely secure. By default Montage will assume the password is “montage”. If you select a different password, make sure to note the password, since we will need to configure Montage with the non-default password. Once we have entered the “Password” and the “Password (again)”, click the “Ok” button to create the user:
We should now see the “montage” user under the Login Roles tree:
Now we create the “montage” database by right clicking the “Databases” tree and selecting “New Database...”: 
On the default “Properties” tab, we enter the “Name” as “montage” and set the owner to our new “montage” Login Role:
Go to the “Definition” tab and ensure that the encoding is “UTF8” (it should already be set to UTF8, but Montage will not correctly work if the encoding is anything else. We can then click “Ok” to create the database, which will take several seconds to allocate the database files:
We should now see the “montage” database (with a red X):
The red X indicates that we have not connected to that database, but double clicking on the database name, will automatically connect and show us the contents of the database (it will currently not have any tables, since the Montage installer has not yet been run):
4.2.2 Montage Settings & License

During the Montage installer, we will be prompted for the Montage installation path, by default C:\montage (in most cases, we should not change this location). Before we start the Montage installer, we must create a subfolder, C:\montage\config, and create two files within that folder.

Create C:\montage:
Create C:\montage\config
Montage Healthcare Solutions issues a unique license for each customer. The license file, `montage-modules.json`, includes the customer’s name, a list of “modules” which that customer has purchased access to (at the current time, Montage ships all production-ready modules to all customers), and potentially a license expiration date. The license file includes cryptographic hashes to prevent manual editing.

The file looks something like:

```json
{
  "expires": "2011-11-01",
  "format": "1.0",
  "modules": {
    "montage_accounts": "XXXXX",
    "montage_analytics": "XXXXX",
    ...
  },
  "organization": "Example Customer",
  "signature": "XXXXX"
}
```
**Warning:** Do not edit the license file, doing so will likely invalidate the license file. If any changes to the license, contact Montage Healthcare Solutions to issue a new license file. Even editing the customer name will cause license invalidation.

Modifying the `SECRET_KEY` in `settings.py` can also invalidate the license.

As part of the *Pre-Install Checklist*, you should have requested from Montage Healthcare Solutions, a unique license file and `SECRET_KEY` for the customer. Place the `montage-modules.json` in the config directory (e.g. at `C:\montage\config\montage-modules.json`).

![Image of config folder]

Ensure that the file extension is `.json` and not `.txt` (in case you used Notepad to save the license file).

**settings.py - Main Config File**

Inside the config folder we also store the `settings.py`, which contains the database connection information, the customer’s `SECRET_KEY`, crawling settings, and various other customer-specific settings. See *Settings* for a more comprehensive guide on what `settings.py` can contain.

To get the installation process started, `settings.py` should contain a `SECRET_KEY`, a basic definition of an index, and the *email settings*:

```python
from montage_basesite.windows_settings import *

SECRET_KEY = 'XXXXX'
MONTAGE_NUANCE_AUTH_KEY = 'YYYYY'
```

---

Chapter 4. Windows Installation Guide
MONTAGE_INDEXES += {
    ('montage_search.index.default.RadiologyIndex', {
        'name': 'rad',
        'display_name': 'Radiology',
        'source_systems': ['Powerscribe360']
    }},
}

from celery.schedules import crontab
from datetime import timedelta
import urllib

CELERYBEAT_SCHEDULE['ps360'] = {
    'task': 'montage_importer.tasks.import_recent_reports',
    'schedule': crontab(hour=1, minute=0),
    'kwargs': {
        'importer': 'montage_importer.importers.vendor.nuance.Powerscribe360',
        'params': {
            'source_system': 'Powerscribe360',
            'db_url': 'mssql+pyodbc:///?odbc_connect=' + urllib.quote_plus('DRIVER={SQL Server};SERVER=10.0.0.50, 1433;DATABASE=Comm4;UID=user;PWD=password'),
            'timezone': 'US/Eastern'
        },
        'offset': timedelta(days=2),
    },
}

A basic description of this example:

- The initial from ... import * line uses an existing settings template that allows us to only define the minimal set of configuration values.

- The SECRET_KEY is used to check the license file and to secure content in the Montage application. You should obtain the SECRET_KEY from Montage Healthcare Solutions.

- The MONTAGE_NUANCE_AUTH_KEY is used to auto-login users from PowerScribe 360 into Montage. This key is the same one used in the RadPortal as the “Montage Secret Key”

- The MONTAGE_INDEXES defines a single Index, which is connected to. “Powerscribe360” Source System. The name and display_name govern what this Index is called within the Montage webpage. The source_systems is a mapping to all the systems that are crawled by Montage, which make up this specific index. In general, this definition of MONTAGE_INDEXES is sufficient. Note the name of the source system, “Powerscribe360” in this case—it must match our crawler definition later.

- In this example, we are configuring the Nuance PowerScribe360 crawler to run every day at 1:00 AM. Refer to General Crawler Configuration for specific instructions on setting up the CELERYBEAT_SCHEDULE. Make sure to configure the correct timezone for the given customer.

If a non-default PostgreSQL database / user / password were set in Create the PostgreSQL User & Database, they must be configured now. Likewise, if using a non-standard data directory, it must be configured.
Ensure that the file extension is `.py` and not `.txt` (in case you used Notepad to save the file):
C:\montage\config should appear so:
4.2.3 Windows Firewall

Montage requires that clients be able to access it via the standard HTTP port 80 (and port 443 if using HTTPS). The customer’s network must have any firewalls set to allow this. Additionally, we set Window’s Firewall to allow Inbound traffic to Apache via port 80 (and if needed 443). Start → Control Panel → Check firewall status → Advanced settings

Click “Inbound Rules” in the left pane:
Click “New Rule...” in the right pane:
Select “Port” as the Rule Type and click “Next”: 
Select TCP and enter “80, 443” (only “80” if HTTPS is not used). Click “Next”: 
Select “Allow the connection”, click “Next”: 
Select “Domain”, “Private”, and “Public”, click “Next”:
The name of the rule is not important, but we can call it “Montage HTTP”. Click “Finish”: 
We can now see our new Inbound rule. We can close the Firewall settings:
4.2.4 Next Steps

We can now proceed onto the Montage Installation.

4.3 Montage Installation

4.3.1 Montage Installer

Having finished the Dependency Installation and Initial Configuration, we are ready to run the Montage installer, montage-VERSION.exe.

Unless we are upgrading, we should install both the Montage Server and the Searchd Server. Click “Install”: 
The installer will take a few minutes:
A command window will open to run `montage-setup.bat`, which creates the tables in the montage database, generates `config\searchd.conf`, and loads static data into the database (like RadLex terms). Do not press any keys, let the batch file complete and automatically close (it may take several minutes to complete):
When completed, click “Close”:
The installer will create several directories in the base install directory:
Directory Structure

- C:\montage
  - lib - The libraries that run Montage
  - searchd-VERSION - The binaries to run the Searchd server
  - montage-VERSION - The Montage Server files (HTML, Python, etc)
  - data - Data files (this directory can live on another partition, but requires customizing file locations in the settings)
  - cache - Montage heavily caches data to avoid hitting slower network resources (like the database). Cached fragments sit in this folder.
  - index - Files created by the Indexer, which Searchd loads and provides results from
  - logs - Application logs, includes usage, audit events, and errors. See Montage Log Files.
  - media - Static files uploaded by users to the site
  - static - Montage combines and minifies static resources like JS and CSS files, it stores the mini-fied resources in this folder
Installed Services

The Montage installer will create 4 new Windows Services.

- “Montage Searchd” is a server which provides quick access to the index files.
- “Montage Celeryd” is an endpoint for the task queue, RabbitMQ. Essentially, it is the worker that takes tasks on the queue and executes them asynchronously.
- “Montage Celerybeat” is a specialized version of “Montage Celeryd” that runs periodic tasks.
- “Montage MLLP Server” is an HL7 receiver, turned off by default

To verify the service are running, open the Services tool via Start → Administrative Tools → Services.

Indexing Scheduled Task

The Montage installer will create 1 new Scheduled Tasks, “Montage Indexer”, which, by default is set to run at 2AM daily. It will regenerate the index files from the most recent crawler run.

**Note:** In future versions of Montage, the Montage Celeryd and Celerybeat tasks will be turned into Services. Currently, the tasks simply run the appropriate programs on boot.

4.3.2 Next Steps

The Montage installer is complete, we can continue onto the Post Installation Configuration to perform the final installation steps and to start the initial data load.

4.4 Post Installation Configuration

4.4.1 Build the Index Files

In order for the “Montage Searchd” Service to work, the index files must be present. These files are built via the nightly “Montage Indexer” Scheduled Task. To avoid waiting until the next nightly run of the Montage Indexer, we can manually run the Scheduled Task: Start → Administrative Tools → Task Scheduler → Task Scheduler Library

Click on the “Montage Indexer” task. It should be in the “Ready” state. In the right bar, click “Run”.

80 Chapter 4. Windows Installation Guide
If we click “Refresh” the task will briefly be in the “Running” state. Wait for the task to return to the “Ready” state (for the initial index file build, when no data has been crawled, this should only take several seconds, once the crawl is finished this can take 20 minutes to an hour). The state is not automatically refreshed, so we may need to repeatedly click “Refresh”.

Once the Montage Indexer task is finished (task in “Ready” state and files present in `<drive>:\montage\data\index\`), we can proceed with turning on the services.

### 4.4.2 Start Services

We now have Montage installed, but we still need to link it to the Apache webserver, so HTTP requests get routed into the Montage Server.

To apply the Apache changes that occurred during the Montage installation, we must restart Apache via the Services interface: Start → Administrative Tools → Services.
At this point, we should check that the “Montage Searchd”, “Montage Celeryd”, and “Montage Celerybeat” services are running. If they are stopped, we start them. We only should start “Montage MLLP Server” if Montage is receiving HL7 messages.
Note: Whenever there is a configuration change (e.g. to C:\montage\config\settings.py), the services ("Apache 2.2", "Montage Celeryd", "Montage Celerybeat", and optionally "Montage MLLP Server") must be restarted to reload the configuration file.

### 4.4.3 Web Configuration

The Montage server features a web-based process for creating the initial admin account and configuring several vital settings. A number of checks will also be run as part of this process to identify common installation problems.

Using the domain we obtained as part of the Pre-Install Checklist, go the URL http://<site’s domain>/install/:
Click the “Start” link. If no admin user has been previous created, we are prompted to create this user. Once the account information is completed, click “Create User”:
We will likely see the Montage EULA. Read the agreement, select “I accept the agreement”, and click “Submit”: 
We must tell Montage its domain, so that it can properly create links. Enter the correct domain name (we obtained as part of the Pre-Install Checklist) in the “Montage Domain” and click “Save”.

Montage will run through a series of checks (it may take several minutes), to ensure the required services are running and that Montage is properly configured. If any checks result in errors or warnings, please follow the instructions or work with Montage Support to resolve the issues:

4.4. Post Installation Configuration
4.4.4 Next Steps

We have configured the base application at this point, the next step is to perform a historical crawl Initial Crawl.

4.5 Initial Crawl

The Montage Celerybeat service will perform daily crawls of any new or updated reports, but we must manually start the process to crawl historical data from PowerScribe 360.

Warning: The initial crawl can be taxing on the upstream database server. Ideally the crawl should be against a mirrored copy of the database, during off-hours, or over a limited time window (using successive runs with --start-date and --end-date). Normal operations will still be possible, but may be slower than normal due to the increased network and CPU load.

Open a command prompt (cmd.exe), change into the bin directory of the Montage installation (C:\montage\lib\montage-VERSION\bin), e.g.:
If we have properly configured the crawler in the CELERYBEAT_SCHEDULE, we can perform the initial crawl via the `montage run_importer` command.

### 4.5.1 Trial Crawl

It is recommended to run a small, trial crawl against a month or two of data:

```
montage.exe run_importer --start-date="2012-05-01" --end-date="2012-06-30" ps360
```

If that succeeds without issue (check `montage-importer.log`), we can proceed to a full crawl.

### 4.5.2 Full Crawl

To run the full crawl (to run in smaller batches, remove the `--full-import` and substitute with `--start-date` and `--end-date`, working your way back in time), execute:

```
montage.exe run_importer --full-import ps360
```
The command runs the `ps360` crawler, as defined in the `CELERYBEAT_SCHEDULE`. The `--full-import` flag indicates that the typical "only grab reports that are new or changed since the last crawl" algorithm does not apply. Instead, all reports should be crawled.

We can monitor the import progress in the looking at `montage-importer.log`. The crawl speed depends upon the network connection to the SQL Server and the processing power of the Montage server. 10k reports typically only take several minutes. Importing a whole RIS can take a day or more.

When completed, we are notified:
While we could wait for the “Montage Indexer” Scheduled Task to run at night, we force the indexer to run by kicking off the Montage Indexer Scheduled Task. To access the file, navigate to C:\montage\lib\searchd-VERSION\ directory.

4.5.3 Installation Finished

At this point, we should now have a fully functioning Montage installation. We can begin using and testing the web site, as well as creating user accounts for other users (must match usernames in PowerScribe 360 for auto-login to work).

4.6 Windows Frequently Asked Questions

Also see Frequently Asked Questions.

- Can Montage be installed on a separate partition?
- The disk has become highly fragmented / Montage is running slow
4.6.1 Can Montage be installed on a separate partition?

Yes, but please contact Montage Healthcare Solutions to properly configure using the separate partition. Currently, the installation should still reside on the C:\ drive, but the PostgreSQL data files and the index files (the primary consumers of drive space) can live on the separate partition using the correct File Locations.

4.6.2 The disk has become highly fragmented / Montage is running slow

Due to the design of the Windows filesystem and the method in which Montage updates reports and performs re-indexing, it is has been observed at some sites that a high level of disk fragmentation can occur.

Since the Windows OS and hardware is managed by the customer’s IT and since for certain types of hard drives, defragmenting can cause unintended drive wear (e.g. solid state drives), we leave it up to the customer to monitor and perform periodic disk degragmenting.

Using the builtin Disk Defragmenter in Windows, you can check the fragmentation level and set up a scheduled defrag (perhaps weekly during off-peak hours).

4.7 Troubleshooting

4.7.1 Tools for Investigating

Log Files

The Montage Log Files are an excellent source to start debugging, specifically, montage.log and the Apache error log contain tracebacks of errors. The montage.log file contains errors inside Montage, while the Apache montage-error.log contains errors that prevent Montage from starting.

4.7.2 Common Problems

When I navigate to the web site, I receive a page with black text on a white background, stating “Not Found” or “Server Error”:

Check the Apache Logs log file for details, the Montage application is not being loaded by Apache.

As soon as I log in, I get a “Server Error” page Most likely, the searchd service not running. Ensure the “Montage Searchd” service is running in Start → Administrative Tools → Services. If it is not running, start the service. If it fails to start, look into the logs (searchd.log and montage.log).

You may attempt to directly launch searchd process to detect the exact error message, for example:

C:\montage\lib\searchd-2.0.1-beta\searchd.exe --config C:\montage\config\searchd.conf

When users receive Password Reset emails, the reset link is incorrect, possible pointing to the example.com domain.

The Web Configuration is not properly configured. Ensure that it points to the correct domain, as obtained in the Pre-Install Checklist

We have updated settings.py, but the changes do not appear to be working The Apache service must be restarted to apply changes, via Start → Administrative Tools → Services. After restarting, if the changes still do not appear, check montage.log for errors.

When I execute montage run_importer I get the following error: (Error) (‘08001’, ’[08001] [Microsoft][ODBC SQL Server Driver][DBNETLIB]Invalidconnection.

(14) (SQLDriverConnectW); [01000] [Microsoft][ODBC SQL Server Driver][DBNETLIB]ConnectionOpen (Invalid Instance()). (14’) None None
This problem occurs due to the ODBC library trying to dynamically determine the SQL Server port (typically port 1433). Consider adding the port, separated by a comma from the server host or IP in the `db_url`, e.g.:

```
db_url': 'mssql+pyodbc:///?odbc_connect=' + urllib.quote_plus('DRIVER={SQL Server};SERVER=192.168.42.178, 1433;DATABASE=Comm4;UID=montage;PWD=montage'),
```

**Installing PostgreSQL fails with “An error occurred executing the Microsoft VC++ runtime installer”.** Likely, the VC++ Runtime was previously installed. We must pass a flag to the PostgreSQL installer to not attempt to re-install this runtime. From the command line, run the PostgreSQL installer with the `--install_runtimes 0` flag:

```
postgresql-9.1.1-1-windows.exe --install_runtimes 0
```

**Installing Python fails with “Another installation is in progress. You must complete that installation before continuing this one.”** Close the Python installer and check for another “msiexec.exe” in the Task Manager running processes (taskmgr.exe). If you see, it the only solution (KB 236456) is to reboot the server and retry the Python installer.

---

4.7. Troubleshooting
SOURCE SYSTEM CRAWLERS

Montage provides a crawler framework, which contains specific adapters for various vendors’ systems and for both standard and custom protocols.

5.1 General Crawler Configuration

Crawlers are defined by adding tasks the the CELERYBEAT_SCHEDULE in settings.py (also see CELERYBEAT_SCHEDULE).

- Template
- Parameters
  - Importer Name
  - task
  - schedule
  - importer
  - offset
  - params
    - source_system
    - db_url
    - timezone
    - detect_qc
    - crawl_radimetrics
    - import_modality
    - exam_codes_are_unique
    - ignore_patient_geo
- Example
- Crawler-specific Options

5.1.1 Template

In settings.py:

```python
from celery.schedules import crontab
CELERYBEAT_SCHEDULE[<crawler name>] = {
    'task': 'montage_importer.tasks.import_recent_reports',
    'schedule': crontab(hour=1, minute=0),
    'kwargs': {
        'importer': <importer class>,
```
5.1.2 Parameters

Importer Name

The crawler name is the key of the CELERYBEAT_SCHEDULE dictionary. It can be any string value. It is used when performing the initial data load to identify which crawler is to be run. For example “ris-import”:

```python
CELERYBEAT_SCHEDULE[‘ris-import’] = {
    ...
}
```

**task**

Celerybeat is a generic framework for executing scheduled tasks, when defining a crawler task, we always use the montage_importer.tasks.import_recent_reports task.

**schedule**

The schedule on which to run the task. Should be a crontab instance.

The crontab can look like:

<table>
<thead>
<tr>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>crontab(minute=&quot;*/15&quot;)</code></td>
<td>Execute every 15 minutes</td>
</tr>
<tr>
<td><code>crontab(minute=0, hour=0)</code></td>
<td>Execute everyday at midnight</td>
</tr>
</tbody>
</table>

The crontab class must be imported prior to the CELERYBEAT_SCHEDULE:

```python
from celery.schedules import crontab
```

**importer**

The class name of the specific crawler adapter we will use to connect to the Source System, each vendor typically requires its own adapter.
offset

The crawler, by default, will crawl only the reports that have been modified since the last successful crawl. We can add the offset parameter, to expand that range to, for example, one days before the last successful crawl, just to make sure we get everything.

The offset parameter is more useful in systems that lack “last modified” timestamp. However, most source systems provide a last modified timestamp that the crawler will automatically use.

The offset value must be a datetime.timedelta

params

A dictionary of parameters that get passed to importer when the class is initialized.

source_system

The name we give to the Source System, this name must match the index definition in MONTAGE_INDEXES. If this source_system does not currently exist in the Montage database, it will be created (therefore, the source system should likely not be renamed after reports have already been crawled from the source system).

db_url

The connection string for the Source System’s database. See Database URL for specific details. The type of connection string will depend upon which specific importer is used.

timezone

Internally, Montage stores all dates in UTC, converting on-demand to the user’s preferred timezone. However, most Source Systems store dates in the local time, e.g. dates in PowerScribe 360 at a hospital in Philadelphia, PA, are assumed to be in the US/Eastern local time. In order to store as UTC in Montage, each crawler must define what timezone the source system stores the dates in. Almost always it is the local timezone. This option is not required if the source system’s database stores dates as timezone-aware. However, in practice, almost no source system stores timezone-aware dates, only timezone-naive dates. Therefore, the timezone parameter is almost always required. Valid timezones include:

- UTC
- US/Eastern
- US/Central
- US/Mountain
- US/Arizona
- US/Pacific
- US/Alaska
- US/Hawaii

5.1. General Crawler Configuration
detect_qc

For each report that is imported, run the QC detection algorithms on it to find QC issues. Detection occurs via the task queue asynchronously.

The value defaults to `False`, since Montage QC is being gradually implemented at sites and since it is only available for Radiology indexes. To enable QC detection, the value must be set to `True`.

crawl_radimetrics

For each report that is imported, look up the report via the Radimetrics eXposure API. Lookup occurs asynchronously via the task queue.

The value defaults to `False`, since many sites do not have Radimetrics and Montage, and since it will only work for Radiology indexes. To enable crawling of the eXposure API, the value must be set to `True`. The appropriate values in settings.py must also be in place to enable the integration (e.g. eXposure URL and credentials).

import_modality

New in version 2.3. For many crawlers, Montage will attempt to obtain a mapping from Exam Types to Modality from the source system. If this mapping is not preferred and the site wishes to manage the mapping manually in Montage, this value can be set to “off”.

exam_codes_are_unique

New in version 2.4.5. For a few customers, multiple exam types may share the same code (this is rare). In this case, Montage can treat the procedures as separate by making the code and description unique together, when this value is set to `False`. The default value, `True`, keeps one Exam Type per code.

A downside of setting to `False` is that the Exam Type will no longer receive updates from upstream on spelling changes to the exam description.

ignore_patient_geo

New in version 2.4. Some crawlers (not all) are able to pull geographic data about patients, such as Zip code, Address, City, & State, which is used in Montage’s Geoanalytics. If the crawler can pull this information, but the customer does not to use this information, this value can be set to `True`.

5.1.3 Example

Below is a full example, which crawls a PowerScribe 360 database every night at 1:00 AM. For the connection string, `db_url`, we connect to the “Comm4” database on the SQL Server at 192.168.42.178, using the “montage” account, which has the password “montage”. (For more details, see ODBC connection string)

Based upon the customer location, we set the timezone to US/Eastern. We are calling the Source System “PowerScribe360” and we have named this specific crawler “ps360”.

```python
from celery.schedules import crontab
from datetime import timedelta
import urllib

CELERYBEAT_SCHEDULE['ps360'] = {
    'task': 'montage_importer.tasks.import_recent_reports',
    'schedule': crontab(hour=1, minute=0),
    'args': (db_url,)
}
```

Chapter 5. Source System Crawlers
5.1.4 Crawler-specific Options

See crawler documentation for specific additional options:

- **Nuance**

5.2 Database URL

5.2.1 ODBC Connection String

```python
import urllib
...
'db_url': 'mssql+pyodbc:///?odbc_connect=' + urllib.quote_plus('DRIVER={SQL Server};SERVER=server;DATABASE=database;UID=user;PWD=password'),
```

5.2.2 MySQL Connection String

...  

5.3 Nuance

The Nuance crawlers make use of an **ODBC connection** to crawl the Microsoft SQL Server.

- Crawler Compatibility
- PowerScribe 360
  - Additional Parameters
    - patient_status
    - use_site_location_as_organization
    - limited_site_names
    - excluded_exam_codes
    - overwrite_on_imported_addendum
- RadWhere
- PowerScribe
5.3.1 Crawler Compatibility

<table>
<thead>
<tr>
<th>Montage</th>
<th>PS 4.x</th>
<th>PS 5.x</th>
<th>RadWhere</th>
<th>PS360 1.x</th>
<th>PS360 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td></td>
<td>y</td>
<td></td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td>y</td>
<td></td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>2.3</td>
<td></td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>2.4</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>3.0+</td>
<td></td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>

5.3.2 PowerScribe 360

Importer Class: `montage_importer.importers.vendor.nuance.Powerscribe360`

Example configuration:

```python
from celery.schedules import crontab
from datetime import timedelta
import urllib

CELERYBEAT_SCHEDULE['ps360'] = {
    'task': 'montage_importer.tasks.import_recent_reports',
    'schedule': crontab(hour=1, minute=0),
    'kwargs': {
        'importer': 'montage_importer.importers.vendor.nuance.Powerscribe360',
        'params': {
            'source_system': 'Powerscribe360',
            'db_url': 'mssql+pyodbc:///?odbc_connect=' + urllib.quote_plus('DRIVER={SQL Server};SERVER=192.168.42.178;DATABASE=Comm4;UID=montage;PWD=montage'),
            'timezone': 'US/Eastern'
        },
        'offset': timedelta(days=1),
    },
}
```

Additional Parameters

This additional parameters can be configured in a `['params']['powerscribe']` dictionary:

```python
CELERYBEAT_SCHEDULE['ps360'] = {
    ...,
    'kwargs': {
        'importer': 'montage_importer.importers.vendor.nuance.Powerscribe360',
        'params': {
            'source_system': 'Powerscribe360',
            ...,
            'powerscribe': {
                'use_site_location_as_organization': <True|False (default)>,
                'patient_status': <mapping dict>,
            },
        },
    },
}
```

1PowerScribe 4.x and 5.x are supported via HL7
patient_status

Defaults to None, but can be configured with a dict of mappings into the I/E/O structure:

```python
'patient_status': {
    'C': 'I',
    'P': 'O',
},
```

use_site_location_as_organization

If set to True, pull the Organization from SiteLocation as opposed to Site.

limited_site_names

New in version 2.4. If set to a list of names from the Site.Name data in PS360 (e.g. ['Baptist', 'University']), the crawler query will only pull information from these sites.

excluded_exam_codes

New in version 2.5.

overwrite_on_imported_addendum

New in version 2.4. If set to True, Montage will only use the Addendum text when constructing the full Report Text when Report.IsAddendum is true. This is useful mainly when historical HL7 imported into PS360 did not split original report text from addendum text.

5.3.3 RadWhere

Importer Class: `montage_importer.importers.vendor.nuance.RadWhere`

Example configuration:

```python
from celery.schedules import crontab
from datetime import timedelta
import urllib

CELERYBEAT_SCHEDULE['radwhere-import'] = {
    'task': 'montage_importer.tasks.import_recent_reports',
    'schedule': crontab(hour=1, minute=0),
    'kwargs': {
        'importer': 'montage_importer.importers.vendor.nuance.RadWhere',
        'params': {
            'source_system': 'RadWhere',
            'db_url': 'mssql+pyodbc:///?odbc_connect=' + urllib.quote_plus('DRIVER={SQL Server};SERVER=192.168.42.178;DATABASE=Comm4;UID=montage;PWD=montage'),
            'timezone': 'US/Eastern',
        },
        'offset': timedelta(days=1),
    },
}
```
5.3.4 PowerScribe

PowerScribe differs significantly from RadWhere and PowerScribe 360. At this time Montage does not provide a crawler for PowerScribe. However, if interest exists, Montage Healthcare Solutions can develop a crawler for PowerScribe.

5.4 Montage HL7 Interface Specification

Montage can be configured with an HL7 listener based on the Minimal Lower Level Protocol (MLLP), consuming ORU^R01 messages that contain the report text and associated metadata. By default, this HL7 listener consumes data on port 6661.

*Montage Support* must be involved to assess feasibility, enable the HL7 listener, and validate the data. Montage Support should be provided with an accurate sampling of de-identified HL7 messages to ensure proper mapping.

The following tables describe the fields and events that Montage is able to consume. Data must be sent according to this specification. Any deviation, may not be supported (contact *Montage Support*).

5.4.1 HL7 Fields

Note: Fields represented in **bold** are required by Montage.
### 5.4.2 HL7 Reporting Events

Montage will consume updates to reports via HL7 according to OBR-25. As the HL7 message is received, Montage will use OBR-25 to determine whether it represents a Preliminary event, a Finalized event, or an Addendum Event (see Result Status). Based upon the Result Status value, Montage will associate the Timestamp (OBR-22) and Provider (OBR-32) with this event, as detailed in Reporting Event.

During this process, Montage will update the fields above and the report text with the latest available values.

For example, if a resident authors a preliminary report (first HL7 message, OBR-25 is “P”), Montage will store the report as it exists, with the resident in OBR-32 as the preliminary author and OBR-22 as the preliminary time. Then, once an attending signs the report (a second HL7 message, OBR-25 is “F”), Montage will update any relevant values and store the attending (OBR-32) as the finalizing provider and OBR-22 as the finalized time. If an addendum is later issued (third HL7 message, OBR-25 is “C”), Montage will update the report and store the radiologist (OBR-32) as the addendum provider and OBR-22 as the addendum time.

---

2By default we assume that the Ordering ID and Reporting ID numbers do not overlap. For example, Ordering ID 3 must not be a different person than than Reporting ID 3. If this is not the case, Montage Support must adjust the HL7 profile
Result Status

<table>
<thead>
<tr>
<th>Event</th>
<th>Result Status (OBR–25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary Event</td>
<td>“P”</td>
</tr>
<tr>
<td>Finalized Event</td>
<td>“F”</td>
</tr>
<tr>
<td>Addendum Event</td>
<td>“C”</td>
</tr>
</tbody>
</table>

Note: OBR–25 must be set appropriately for all messages.

Reporting Event Definition

<table>
<thead>
<tr>
<th>Field</th>
<th>Segment</th>
<th>Component</th>
<th>Sub-component</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting Timestamp</td>
<td>OBR</td>
<td>22</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Reporting Provider ID</td>
<td>OBR</td>
<td>32</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Reporting Provider Last Name</td>
<td>OBR</td>
<td>32</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Reporting Provider First Name</td>
<td>OBR</td>
<td>32</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

5.5 Montage File Format Extract Specification

Montage can watch a directory for new files and import any files. Montage’s ability to import both structured and un-structured files make it excellent for searching legacy information systems.

5.5.1 File Format Types

Montage can crawl the following file types:

- Delimited Files
  - Comma separated value (CSV)
  - Tab separated value (TSV)
  - Other delimiters, e.g. piped delimited
- XML Files
- Plain Text Files

^[3]Unless otherwise noted, timestamps are expected to be formatted using the notation described in section 2.A.22 of the HL7 V2.5-2003 standard. YYYY[MM][DD][HH][MM][SS][SS][SS][SS][SS]+[/-ZZZZ]
5.5.2 Delimited Files

The files should follow the RFC 4180 specification. This includes:

- Each report must occupy one row in the file
- The first row must be a header.
  - Montage keys off the column names, so it is imperative that these names remain constant over time.
  - Ordering of the columns does not matter.
  - There should be no other header or footer rows
- Fields must be quoted by double quote characters ("), if:
  - The field’s content has internal newlines (i.e. the report text)
  - The field’s content includes the delimiter character (e.g. comma if CSV, tab if TSV, etc.)
  - Fields with internal double quotes, must be escaped by replacing the single double-quote with a pair of double quotes (e.g. he said, "Hello, turns into he said, ""Hello"
- Any delimiter is acceptable, as long as it is properly quoted when that delimiter character appears inside the content, as mentioned above.
- Ideally the file will be in UTF-8, with no Byte Order Mark (BOM).
  - If not in UTF-8, the exact encoding must be communicated to Montage Support.
  - The file should be tested with non-ASCII characters. For example, common non-ASCII characters that have alternate representation depending upon the encoding include the left double quotation mark, " (Unicode: U+201C), and the degree sign ° (Unicode: U+00B0)
- Dates should follow ISO 8601 (e.g. YYYY-MM-DD HH:MM:SS), though Montage can understand most dates Microsoft Excel parses. Unless specified, it is assumed that the dates are in the local timezone.
- Microsoft Excel must be able to open or import the file and the contents must be displayed in the correct columns and rows. A common issue is that internal newlines are not properly quoted

If the file does not meet this specification, it is possible that Montage still may be able to crawl it, but it may take significant additional resources to properly handle ill-formed data.

Note: In the below explains we use \n to represents an ASCII “LF” (line feed) and \r to represent an ASCII “CR” (carriage return), which are non-visual characters. The files must encode line feeds and carriage returns in as actual the ASCII values at both the end of the rows as well when inside the cells’ values (e.g. you should not be able to ever see “rn” when opening up in Excel).

CSV File Format Example:

An example that demonstrates proper field quoting with internal newlines, quotes, and commas a CSV formatted file:

```
Accession,Organization,Exam Code, Exam Description,Exam Date,Report Text\n123,Baptist,CT120,CT of Head,2012-04-26 11:00:32,"History:\nInternal newline\n124,Lutheran,CT120,CT of Head,2012-04-26 11:13:00,"Internal comma, and ""quote"" are escaped\n```

Pipe-Delimited Example:

The following is an example that demonstrates proper field quoting with internal newlines, quotes, and commas of a pipe-delimited file:
Common Problems

The file opens in Microsoft Excel, but the report text is spread across multiple rows.

The report text field is not properly escaped by quoting. Ensure the report text fields start and end with double quotes and that any internal double quotes are escaped by a pair of double quotes.

If the quoting approach is not feasible for some reason, it may be possible (with additional effort), to consume reports where the newline characters have been replaced or escaped by some other means. Contact Montage Support for assistance.

Fields

The following tables list the fields that are either required or suggested to be imported into Montage.

Table 5.1: Required Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facility</td>
<td></td>
</tr>
<tr>
<td>Accession Number</td>
<td></td>
</tr>
<tr>
<td>Report Text</td>
<td></td>
</tr>
<tr>
<td>Procedure Code</td>
<td></td>
</tr>
<tr>
<td>Procedure Description</td>
<td></td>
</tr>
<tr>
<td>Patient MRN</td>
<td></td>
</tr>
<tr>
<td>Exam Completed Timestamp</td>
<td>Format: YYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td>Reporting Provider ID*</td>
<td></td>
</tr>
<tr>
<td>Reporting Provider Last Name</td>
<td></td>
</tr>
<tr>
<td>Reporting Provider First Name</td>
<td></td>
</tr>
<tr>
<td>Finalized/Reporting Timestamp</td>
<td>Format: YYYY-MM-DD HH:MM:SS</td>
</tr>
</tbody>
</table>
Table 5.2: Suggested Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure Modifier</td>
<td></td>
</tr>
<tr>
<td>Modality</td>
<td></td>
</tr>
<tr>
<td>Is Stat?</td>
<td></td>
</tr>
<tr>
<td>Ordering Provider ID*</td>
<td></td>
</tr>
<tr>
<td>Ordering Provider Last Name</td>
<td></td>
</tr>
<tr>
<td>Ordering Provider First Name</td>
<td></td>
</tr>
<tr>
<td>Ordered Timestamp</td>
<td>Format: YYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td>Preliminary Provider ID*</td>
<td></td>
</tr>
<tr>
<td>Preliminary Provider Last Name</td>
<td></td>
</tr>
<tr>
<td>Preliminary Provider First Name</td>
<td></td>
</tr>
<tr>
<td>Preliminary Timestamp</td>
<td>Format: YYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td>Addendum Provider ID*</td>
<td></td>
</tr>
<tr>
<td>Addendum Provider Last Name</td>
<td></td>
</tr>
<tr>
<td>Addendum Provider First Name</td>
<td></td>
</tr>
<tr>
<td>Addendum Timestamp</td>
<td>Format: YYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td>Patient Status</td>
<td>Format: “I”, “O”, or “E”. * If there are others, we can map them into I/O/E.</td>
</tr>
<tr>
<td>Patient Last Name</td>
<td></td>
</tr>
<tr>
<td>Patient First Name</td>
<td></td>
</tr>
<tr>
<td>Patient Date of Birth</td>
<td>Format: YYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td>Patient Sex</td>
<td></td>
</tr>
</tbody>
</table>

5.5.3 XML Files

Custom XML file formats can be consumed. Contact Montage Support.

5.5.4 Plain Text Files

Custom plain text (unstructured) files can sometimes crawled. Contact Montage Support for an analysis.

5.5.5 Other

Contact Montage Support for an effort analysis.

---

**Note:** It is important to note that the Montage crawler is idempotent, meaning reports can safely be re-imported multiple times. Before the crawler inserts the report into the Montage database, it checks to see if the report already exists. If the report already exists, the existing record is updated. So Montage only contains the latest information about a given report.

Montage contains crawlers for a number of additional vendors. Additionally, Montage contains HL7, CSV, and XML crawlers. If a customer is interested in interfacing with another Source System, please contact Montage Healthcare Solutions.
The primary location to configure the Montage Server is in settings.py (typically located in /etc/montage on Linux and C:\montage\config on Windows). settings.py is a Python source file, which adds a small amount of complexity, but allows us to easily define interesting and complicated configurations. There are several additional settings that are controllable via the web-interface (thus allowing an admin at a customer site to adjust the settings). However the settings in settings.py should only be adjusted by someone knowledge in their impact.

**Note:** On Windows, since Python treats “\” as an escape character, all file and directory paths must use “\\” instead. For example, C:montage should actually be C:\montage.

To apply changes after settings.py is saved, Apache must be restarted. On Linux service apache2 restart will reload the Montage Server. On Windows, restart Apache via the Services tool.

### 6.1 Authentication

See the related *LDAP Settings* section.

#### 6.1.1 MONTAGE_AUTH_INTERNAL_IS_PRIMARY

Default: True

If set to False, hides many of the tools for managing internal passwords. Useful for LDAP-only configurations.

#### 6.1.2 MONTAGE_AUTH_ALLOW_RESETS_VIA_EMAIL

Default (on Windows): False Default (on Linux): True

If True, users can request and receive a unique password reset email. If False, a Montage administrator must go into the Admin, click the “Reset Password” button for the user and manually communicate the unique URL to the user.

Requires a valid Email configuration. On most Linux installs, a local Postfix install is present and automatically configured (though using an external SMTP server is still recommended). On all Windows installs, no mail server is present and an external EMAIL_HOST (and related settings) must be configured.
6.1.3 MONTAGE_AUTH_ALLOW_PASSWORD_CHANGE

Default: True

If set to False, users can not change their internal password. Useful for LDAP-only authentication configurations.

6.2 Crawlers

Crawlers allow Montage to extract data from external systems. See General Crawler Configuration for details on configuring crawlers.

6.3 Email

On Windows, email is disabled by default. On Linux, Montage will try to use a local SMTP mailer (though this method can often get marked as spam). Therefore, it is recommended if wishing to use password resets or other outbound email features of Montage, an appropriate EMAIL_HOST should be configured with valid local email addresses in SERVER_EMAIL and DEFAULT_FROM_EMAIL.

6.3.1 SERVER_EMAIL & DEFAULT_FROM_EMAIL

Default: ‘admin@montagehealthcare.com’

DEFAULT_FROM_EMAIL is the From address for emails sent to users, SERVER_EMAIL is the From address for error emails sent to ADMINS. Both of these settings should be changed to a valid, customer-specific address.

The email address can be the same for both settings.

**Warning:** If the default is not changed to a valid, local email address, it is likely that users’ email clients will mark any email coming from the Montage server as spam, since the montagehealthcare.com domain has a specific defined list of valid SMTP servers that may send email on its behalf.

6.3.2 EMAIL_HOST

Default: ‘localhost’

The host to use for sending email.

6.3.3 EMAIL_HOST_PASSWORD

Default: ” (Empty string)

Password to use for the SMTP server defined in EMAIL_HOST. This setting is used in conjunction with EMAIL_HOST_USER when authenticating to the SMTP server. If either of these settings is empty, Montage won’t attempt authentication.

6.3.4 EMAIL_HOST_USER

Default: ” (Empty string)

Username to use for the SMTP server defined in EMAIL_HOST. If empty, Montage won’t attempt authentication.
6.3.5 EMAIL_PORT

Default: 25
Port to use for the SMTP server defined in EMAIL_HOST.

6.3.6 EMAIL_USE_TLS

Default: False
Whether to use a TLS (secure) connection when talking to the SMTP server.

6.3.7 EMAIL_SUBJECT_PREFIX

Default: '[Montage] '
The prefix for email messages coming from the Montage server.

6.3.8 ADMINS

Default: () (Empty tuple)
A tuple that lists people who get code error notifications. Montage will email these people with the full exception information. Each member of the tuple should be a tuple of (Full name, email address). Example:

```python
ADMINS = (('John', 'john@example.com'), ('Mary', 'mary@example.com'))
```

6.4 External Launchers

Provide buttons to launch PACS or other web-based viewers for each report.

Requires the site has a PACs that can accept some context (typically the report’s accession number) as a URL parameter to launch the PACS.

Montage ships with a number of built-in launchers, including the ability to check permissions and perform audit logging. Additionally, custom launcher logic can be defined, for instance if at a multi-organization site, multiple PACS exists, holding different subsets of images.

For all built-in launchers, the Can use external launchers (e.g. PACS) permission will be checked. If a user or group lacks this permission, the user will not see the button to view images.

6.4.1 MONTAGE_EXTERNAL_LAUNCHERS

New in version 2.5. Default: None
Takes a list of launchers and the parameters those launchers require. Each report can have multiple launchers (new in 2.5). Each launcher is configured with a list of source_systems it should be shown for. If a report is from that source system, the button will be displayed.

The launchers are typically configured with a url template that can make use of the Available Context to interpolate data such as the accession number into a URL to direct the browser to.

See further below for a full example, but the general pattern looks like:
MONTAGE_EXTERNAL_LAUNCHERS = [
    {'montage_external_launcher.launchers.SimpleLauncher', {'
        'label': 'View Images',
        'url': 'https://pacs.hospital.org?acc={report.accession_number}',
        'source_systems': ['Powerscribe360'],
        'title': 'Open this study in the web PACS viewer',
    }}],

Note, prior to Montage 2.5, launchers were defined per-index (typically as def external_launcher() in indexes.py). As of 2.5, this approach is deprecated and will be removed in an upcoming release.

### 6.4.2 Launchers

**montage_external_launcher.launchers.SimpleLauncher**

The SimpleLauncher puts a direct link to the viewer system, much like the pre-2.5 style external_launcher. The link will open a new browser window to the image viewer.

**Required Parameters:**

- `url` - Launch URL template, that can use the Available Context. For example, 'https://pacs.hospital.org?acc={report.accession_number}'
- `label` - Label on the button (e.g. 'View Images')
- `source_systems` - a list of Source System that the launcher should apply to.

**Optional Parameters**

- `window_name` - Name used by window.open. Default: 'webpacs'
- `window_features` - Features passed to window.open
- `title` - Message that is displayed when hovering above the button

**montage_external_launcher.launchers.RedirectLauncher**

Similar to the SimpleLauncher, but indirectly links to the viewing system, by hitting the Montage server again and using a HTTP Redirect to bounce over to the viewing system. This approach “hides” the URL context a bit better than the SimpleLauncher (e.g. if a username / password must be placed in the url). The RedirectLauncher also audit logs any time a user clicks on the launcher button.

Takes the sample Required and Optional Parameters as the SimpleLauncher, but additionally requires:

- `slug` - A string used to uniquely identify this launcher. If multiple launchers are configured, they each must have a different slug.

**Custom Launchers**

Custom launchers can be defined that have different launching rules or approaches. Contact Montage Support for assistance.

Typically this involves defining a custom subclass of BaseLauncher or SimpleLauncher and overriding the render() or get_render_context() methods.
6.4.3 Available Context

Two variables are typically available in the context used to define the `url`: `report` and `user`. Each of these variables have attributes that can be accessed in the template.

- **report**
  - accession_number
  - patient_mrn
- **user**
  - username
  - first_name
  - last_name

For example, to pass the accession number and current username:

```python
'url': 'https://pacs.hospital.org?acc={report.accession_number}&user={user.username}'
```

6.4.4 Examples

An example using that will add two launchers for the IDXRad system. One is a direct SimpleLauncher and one is an audited RedirectLauncher:

```python
MONTAGE_EXTERNAL_LAUNCHERS = [
    ('montage_external_launcher.launchers.SimpleLauncher', {
        'label': 'View Images',
        'url': 'https://pacs.hospital.org?acc={report.accession_number}',
        'source_systems': ['IDXRad'],
        'title': 'Open report in new web PACS window'
    }),
    ('montage_external_launcher.launchers.RedirectLauncher', {
        'slug': 'pacs',
        'label': 'View Images',
        'url': 'https://pacs2.hospital.org?acc={report.accession_number}',
        'source_systems': ['IDXRad'],
    }),
]
```

6.5 File Locations

**Note:** On Windows, since Python treats “\” as an escape character, all file and directory paths must use “\\” instead. For example, `C:\montage` should actually be `C:\\montage`

6.5.1 MEDIA_ROOT

Linux Default: `/var/lib/montage/static`

Windows Default: `C:\montage\data\static`
Location where uploaded files are stored. It is served by Apache at /media/

6.5.2 STATIC_ROOT

Linux Default: /var/lib/montage/static
Windows Default: C:\montage\data\static

Location where Montage’s static files (CSS, JS, images) are stored. It is served by Apache at /static

6.6 General Settings

6.6.1 SECRET_KEY

Default: ’’ (Empty string)
A secret key for this particular Montage installation. Used to provide a seed in secret-key hashing algorithms and to check the Montage license.

6.6.2 MONTAGE_NUANCE_AUTH_KEY

Default: None
A shared secret between PowerScribe 360 and Montage to allow for auto-logins by PS360 users into Montage. Still requires that the user accounts exist and have the proper permissions in Montage.

6.6.3 DATABASES

Default:

```
DATABASES = {
    'default': {
        'HOST': 'localhost', # PostgreSQL server IP / host
        'PORT': '5432', # PostgreSQL port
        'NAME': 'montage', # database name
        'USER': 'montage', # name of database user
        'PASSWORD': 'montage',
    }
}
```

The connection information for the Montage database connection. Individual settings may be customized:

```
DATABASES[‘default’][‘NAME’] = ‘montage_test’
DATABASES[‘default’][‘PASSWORD’] = ‘SuperSecretPassword’
```

6.6.4 CELERYBEAT_SCHEDULE

A dictionary of the scheduled tasks. Often used for General Crawler Configuration. A number of tasks are shipped as part of the montage_basesite.settings, thus CELERYBEAT should only be added to by individual key:
from celery.schedules import crontab
CELERYBEAT_SCHEDULE["my_task"] = {
    'task': 'path.to.my_task',
    # execute daily, at 1AM local
    'schedule': crontab(hour=1, minute=0),
}

6.6.5 MONTAGE_REQUESTS_PROXIES

New in version 2.4. Default: None (no proxies)

An optional dictionary of HTTP proxies to use when Montage requests external internet sites (e.g. stats.montagehealthcare.com or yottalook.com).

Should use protocol (http, https) as the key and the proxy server as the value:

MONTAGE_REQUESTS_PROXIES = {
    'http': 'http://10.10.1.10:3128',
    'https': 'http://10.10.1.10:3128',
}

Note that the exact values used will depend upon the proxy. Some proxies will use the same protocol and port for both http and https traffic, others may use different values. However, both the http and https keys are expected to be set, since Montage uses both protocols when communicating with external services.

6.6.6 MONTAGE_MAX_EXPORT

Default: 10000

Limit to how many reports can be exported into XLS or CSV. This value can be increased to allow for larger exports, but may impact system performance.

There is currently a 100000 absolute upper bound when exporting from Montage Search. Additionally the XLS file format only supports ~65k rows.

Contact Montage Support for assistance or to express interest in larger export ranges.

6.7 Group Organizations

Montage Group Organizations allows users to only access reports from the Organizations they have been granted permission to view. Useful for multi-hospital installs that wish to limit what data each hospital can view. Permission to view an organization is granted to a group in the Admin.

Warning: If this feature is enabled, extra Admin effort is required to set up and maintain the Organization-to-Group and Group-to-User mappings.
If a user does not see any data in the application, it is likely that her account has not been put into any Groups that have an Organization mapped.

6.7.1 MONTAGE_GROUP_ORGANIZATIONS_ENABLED

New in version 2.5. Default: False
If True, enable the Group Organizations feature.
6.8 Indexes

6.8.1 MONTAGE_INDEXES

Montage can have multiple “indexes” (commonly there is a Radiology index, a Pathology index, and a combined PathRad index). Each index is defined by:

- A name, which is used in the URL, e.g. “http://montage.example.org/search/rad”
- A display_name, which is the full name, e.g. “Radiology”
- A set of Source Systems which contain the reports that comprise the index.

The definition of indexes can get somewhat complex, offering hooks into how the reports are formatted and what image launchers to use. However, at its simplest, we can define the indexes as:

```python
MONTAGE_INDEXES += (
    ('montage_search.index.default.SimpleSourceSystemIndex',
     {'name': 'rad', 'display_name': 'Radiology', 'source_systems': ['Powerscribe360']}),
)
```

In the above example, we define a single index, named ‘rad’, which maps to a single source system, “Powerscribe360”. The source system name must match the name used in the crawler’s source_system.

**Note:** It is important to use the `+=` operator, since we want to append to the existing list of indexes (which includes several “system indexes” such as the spell corrector).

6.9 LDAP / ActiveDirectory

Montage, by default, uses its own secure authentication and authorization system. However, many sites have existing LDAP or ActiveDirectory implementations (henceforth just referred to as LDAP), which Montage can leverage to avoid users managing yet another password.

LDAP can also be useful if there are site-specific requirements of password complexity, aging, etc.

LDAP configuration is complex and error prone (often caused by a lack of proper information given about the LDAP server and LDAP structure). Additionally, most LDAP servers to use a Distinguished Name (DN) for the user instead of the standard “username”, thus requiring a separate step to search for the DN based upon the username before checking the password for that DN. It is recommended to work with Montage Support to test and implement the LDAP configuration. Additionally, there are many configuration options that allow complex configurations to exist (e.g. hybrid systems where some users authenticate with LDAP while others use Montage’s internal authentication).

Below we will give an example of configuring Montage to use Microsoft ActiveDirectory. Montage can use other LDAP providers such as OpenLDAP or Novell, but several parameters may differ. Contact Montage Support for assistance with non-ActiveDirectory configurations.

6.9.1 Information Required from Customer

Before proceeding, several pieces of information are required from the customer:

- A service account to bind to LDAP. We will need the full DN of this account (username may not be sufficient). The DN may look something like `CN=Svc_montage,OU=Service Accounts,DC=HOSPITAL,DC=ORG`. We also need the service account’s password (which ideally should not expire, or it will lock users out)
• The LDAP server URL, e.g. ldap://ad.hospital.org

• The search tree to look for users, e.g. OU=Radiology,DC=HOSPITAL,DC=ORG. All users who log into Montage must somewhere in the subtree under this entry.

### 6.9.2 How Montage’s LDAP integration works

Once configured, the following process occurs when the user comes to the Montage login page and she can enter her username and password.

Montage connects to the LDAP server (AUTH_LDAP_SERVER_URI), binding with AUTH_LDAP_BIND_DN and AUTH_LDAP_BIND_PASSWORD.

Montage then performs an LDAP search AUTH_LDAP_USER_SEARCH to find the user’s DN based upon the user’s username (e.g. we find the DN “CN=Smith, Sally,OU=Users,DC=example,DC=com” for the username “sallysmith”)

Using this DN and the user’s entered password, Montage attempts a second bind operation to check to see if the password is valid.

If the password is not valid, the user gets an “invalid username / password message” and the process stops.

For a valid login, Montage will then check to see if there is an existing user account in Montage for that user. If so, the user will authenticate into that account and get all the permissions for that account and associated groups.

If the login was valid, but no account exists yet for that user, a shell account will be generated that lacks any permissions. The user will get a “unauthorized” message and be unable to use Montage. A local administrator must then go and add this shell account into the “Montage Users” group and assign any extra required permissions. It is therefore recommended that user accounts get set up in Montage with the proper permissions before users attempt to log into Montage to avoid them getting the unauthorized message.

### 6.9.3 Debugging

If issues when user’s try to login to Montage using LDAP, you can enable DEBUG logging into the montage.log. In settings.py`, add the following line and restart Apache:

```python
LOGGING['loggers']['django_auth_ldap']['level'] = 'DEBUG'
```

It is not recommended to run with this setting in production, as it will log several lines every time a user tries to log in.

### 6.9.4 Example Configuration

See the configuration details below for LDAP Settings and Authentication.

Example ActiveDirectory Configuration that disables the internal Montage authentication:

```python
# Turn off the internal authentication
MONTAGE_AUTH_INTERNAL_IS_PRIMARY = False
MONTAGE_AUTH_ALLOW_RESETS_VIA_EMAIL = False
MONTAGE_AUTH_ALLOW_PASSWORD_CHANGE = False

# Configure LDAP
from montage_ldap.ldap_settings import *
AUTH_LDAP_SERVER_URI = 'ldap://ad.hospital.org'
AUTH_LDAP_BIND_DN = 'CN=Svc_montage,OU=Service Accounts,DC=HOSPITAL,DC=ORG'
AUTH_LDAP_BIND_PASSWORD = '<password>'
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    'OU=Radiology,DC=HOSPITAL,DC=ORG',
    AUTH_LDAP_USER_SEARCH,
    ldap.SCOPE_SUBTREE,
    ['objectClass', 'cn', 'uid'], ldap.FILTER_AND,
    )
```

---

**6.9. LDAP / ActiveDirectory** 117
ldap.SCOPE_SUBTREE,
'(sAMAccountName=%(user)s)'
}

AUTH_LDAP_CONNECTION_OPTIONS = {
    ldap.OPT_REFERRALS: 0,
}

For user-simplicity, it is recommended that accounts are still set up in Montage with the correct permissions before users attempt to log in via LDAP, since LDAP only controls authentication, not authorization.

### 6.9.5 LDAP Settings

For LDAP-only setups, it is often recommended to also set `MONTAGE_AUTH_ALLOW_PASSWORD_CHANGE`, `MONTAGE_AUTH_ALLOW_RESETS_VIA_EMAIL`, and `MONTAGE_AUTH_INTERNAL_IS_PRIMARY` to `False`.

For these LDAP settings, must also add an import:

```python
from montage_ldap.ldap_settings import *
```

**AUTH_LDAP_SERVER_URI**

Default: ‘ldap://localhost’

LDAP Server URL, may look something like ‘ldap://ad.hospital.org’

**AUTH_LDAP_START_TLS**

Default: False

If True, use TLS security on the connection to the `AUTH_LDAP_SERVER_URI`

**AUTH_LDAP_BIND_DN**

Default: 

The Distinguished Name (DN) of a Service Account to perform the initial LDAP bind that allows for searching of the user’s DN based upon her username.

The DN typically looks something like `CN=Svc_montage,OU=Service Accounts,DC=HOSPITAL,DC=ORG`.

**AUTH_LDAP_BIND_PASSWORD**

Default: 

The Service Account password to use during initial bind.

**AUTH_LDAP_USER_SEARCH**

A `LDAPSearch` instance that defines the part of the LDAP tree to look under and the filter to use within LDAP entries to find the right user based upon the username.
The search tree is the lowest point in the LDAP search tree which includes all users that will use Montage. For instance if users’ DNs look like ‘CN=Smith\, Sally,OU=Radiology,DC=HOSPITAL,DC=ORG’, we might try to search under ‘OU=Radiology,DC=HOSPITAL,DC=ORG’.

For MS ActiveDirectory, the filter is almost always ‘(sAMAccountName=%(user)s)’.

For example:

```
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    'OU=Radiology,DC=HOSPITAL,DC=ORG',
    ldap.SCOPE_SUBTREE,
    '(sAMAccountName=%(user)s')
)
```

```
AUTH_LDAP_CONNECTION_OPTIONS = {}
```

Extra options to pass when connecting to the LDAP server. One commonly used option is turning off referrals for MS ActiveDirectory (via `ldap.OPT_REFERRALS: 0`). For example:

```
AUTH_LDAP_CONNECTION_OPTIONS = {
    ldap.OPT_REFERRALS: 0,
}
```

### 6.10 Analytics

#### 6.10.1 MONTAGE_ANALYTICS_DISPLAY_TODAY

Default: `False`

If set to `True`, the Analytics Dashboards will display an additional tab with Statistics from the current day. This setting is only useful when data is being crawled or received in near-time, as any data not yet crawled will not be displayed.

#### 6.10.2 MONTAGE_ANALYTICS_MAX_WARM_CACHE_TIME

Default: 7200 (2 hours)

Number of seconds to spend pre-computing the analytics cache after the crawler runs. Two hours is the recommended period to allow pre-computation of the most common graphs.

#### 6.10.3 MONTAGE_RESIDENCY_ANALYTICS

Default: `False`

If set to `True`, the Resident Analytics dashboard will be turned on in Montage Analytics. The site is required to configure its residents via the Admin > Providers > Edit.
6.10.4 MONTAGE_BETA_CUMULATIVE_TAT

Default: False

If set to True, the Cumulative TAT link will be displayed under Montage Analytics.

There are future plans to turn a more full-featured and configurable version of this beta feature on by default.

6.11 Purpose of Use

Montage Purpose of Use requires users to identify why they are using Montage (e.g., research, QA, training). The available purposes are configurable in the Admin. This feature is not enabled by default.

6.11.1 MONTAGE_PURPOSE_ENABLED

New in version 2.3. Default: False

If True, enable the Purpose of Use feature.

6.11.2 MONTAGE_PURPOSE_PER_SESSION

Default: False

If True and MONTAGE_PURPOSE_ENABLED also enabled, users will have to pick a purpose upon every login.

6.12 Radimetrics eXposure

In addition to the required license and crawler settings, to enable the Radimetrics eXposure integration in Montage (crawling, dose analytics, and eXposure launching), 3 parameters must be set (defined below):

- MONTAGE_RADIMETRICS_URL
- MONTAGE_RADIMETRICS_USERNAME
- MONTAGE_RADIMETRICS_PASSWORD

Configuring Radimetrics integration will require a rebuild of the Montage SearchD configuration file as well as a re-index, in addition to a restart of Apache and the Celeryd and Celerybeat services.

Due to the complexity, it is recommended to work with Montage Support when configuring.

A historical load of Radimetrics data will require manual intervention by Montage Support.

6.12.1 MONTAGE_RADIMETRICS_URL

The URL to the eXposure server, including protocol and port (if not default), with no trailing slash. E.g. ‘http://172.16.10.127:8080’

6.12.2 MONTAGE_RADIMETRICS_USERNAME

Credentials to use when authenticating against eXposure.
6.12.3 MONTAGE_RADIMETRICS_PASSWORD

Credentials to use when authenticating against eXposure.

6.12.4 MONTAGE_RADIMETRICS_MRN_RESTRICT

Default: True

Optional setting, if set to False will only query the eXposure API using the accession number.
The enterprise search and business analytics tool for radiology.

7.1 Commands

7.1.1 run_importer

Run a given crawler (identified by its task-name). By default, it will only crawl the new or updated reports (unless --full-import is set).

Usage:

```
```

*task-name* is name of the crawler (i.e. the dictionary key from *CELERBEAT_SCHEDULE*).

Options

--full-import Disregard the new-and-update algorithm, instead crawl the entire source system (WARNING: can be taxing on the database being crawled).

--start-date="YYYY-MM-DD" Import records that were created since the provided data. Typically used in conjunction with --end-date, to successively crawl historical data. Does not make sense to use in conjunction with --full-import.

--end-date="YYYY-MM-DD" Import records that were created before the provided data.

--detect-qc Run the QC detection algorithms on all imported reports. Defaults to off. In most cases, it is recommended to not enable this flag during the historical migration, since prior QC events are less relevant.

--crawl-radimetrics Lookup each imported report in the Radimetrics eXposure API. Defaults to off. Requires proper configuration of the Radimetrics settings to work. In most cases, it is recommended to not enabled this flag during the historical migration.

--disable-signals NOT RECOMMENDED. Turn off posting tasks to the task queue during the import. Several important post-processing tasks occur via the task queue, thus only use if you know how to later run the post processing.

7.1.2 changepassword

**Warning:** Do not use for accounts managed via LDAP/ActiveDirectory, since the changepassword command only works against the local authentication system.
Change the password for a given *username*.
Usage:

```
montage changepassword username
```
BACKING UP MONTAGE

Montage does not currently include an automatic backup utility, so it is left up to the server administrator to perform regular backups (see below for sample backup scripts).

There are four important sets of information that should be backed up:

- The “montage” database in PostgreSQL (includes crawled reports, user settings, etc.). PostgreSQL should be backed up using the `pg_dump` tool or its GUI equivalent in pgAdmin, called “Backup”. These tools will export a SQL file, which can then be restored via the `psql` command or pgAdmin. See below for sample Windows and Linux backup scripts.

- Configuration files, living in `/etc/montage` or `C:\montage\config`.

- Log files (including audit logs). It is advised to backup both the Montage logs and the Apache access logs. On Linux, `/var/log/montage` and `/var/log/apache2`. On Windows, `C:\montage\data\logs` and the wherever the Apache stores its logs (often `C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs`). Note, in `settings.py`, Montage can be configured to store logs elsewhere.

- Uploaded files, which are stored at `/var/lib/montage/media` or `C:\montage\data\media`. The searchd index files `C:\montage\data\index` can safely be ignored, since they are completely rebuilt daily.

Montage can run in a clustered configuration, if high availability is important.

8.1 Windows PostgreSQL Backup

8.1.1 Step 1: Create a Backup Directory

Create a directory where the backup files will be stored, i.e. `C:\MontageBackups`.

8.1.2 Step 2: Create Batch File

Open notepad and copy and paste the following lines (Adjust `BACKUP_FILE` appropriately based upon Step 1):

```bash
set BACKUP_FILE= C:\MontageBackups\montage-db.backup
echo backup file name is %BACKUP_FILE%
SET PGPASSWORD=montage
echo on
"C:\Program Files (x86)\PostgreSQL\9.1\bin\pg_dump" -i -h localhost -p 5432 -U montage -F c -b -v -f %BACKUP_FILE%
```

Save the file with the name, MontageDatabaseBackup.bat (ensure the file extension is .bat, not .txt).
8.1.3 Step 3: Create a Scheduled Task

1. Open Task Scheduler
2. Right-click on Task Scheduler Library
3. Select “Create Task...” option from the popup menu
4. In the General tab, give the task a name, i.e. Montage Database Backup
5. Click on Actions tab.
6. Click on “New...” button.
7. In the New Action window, click on “Browse...” button.
8. Navigate to the directory created in Step 1.
9. Select the batch file created in Step 2.
10. Click on Triggers tab.
11. Click on “New...” button.
12. Setup a new trigger on how and when this task should run.

Note: Test to make sure the Task runs successfully. If the file is not created properly may need to enter full path when setting BACKUP_FILE.

8.1.4 Restoring the Database From the Backup

1. Using pgAdmin rename the original ‘montage’ database
2. Using pgAdmin, create a new ‘montage’ database.
3. Open command prompt and execute (change path as necessary):

   "C:\Program Files (x86)\PostgreSQL\9.1\bin\pg_restore" -h localhost -p 5432 -U montage -W -d montage montage-db.backup

4. When prompted enter ‘montage’ for the password and hit Enter.

8.2 Linux PostgreSQL Backup

To backup, use pg_dump to dump into a gzip’ed SQL file (adjust as desired):

   sudo -u postgres pg_dump -O montage | gzip > montage-db.sql.gz

Since this file is just SQL, we can restore via psql:

   gunzip -c montage-db.sql.gz | psql -U montage -h localhost montage
CHAPTER
NINE

ADDITIONAL MONTAGE TOOLS

9.1 Truncate (Wipe) the Database

**Warning:** This action is **destructive**. Run only if you know exactly what you are doing. *Montage Support* is always available if you want to check or need assistance. A backup is required prior to executing these commands.

If Montage has been set up to crawl a test source system (e.g. Powerscribe 360 was undergoing initial testing and Montage crawled and indexed the test data), and this data needs to be cleared out of Montage prior to going live, the below commands will allow the entire set of reports to be wiped away.

These commands will essentially reset Montage, leaving only User accounts. All data (reports, procedures, organizations, etc.) will be removed. If you require a more precise cleaning, contact *Montage Support* to work out a more localized plan to clean up test data.

9.1.1 Step 0: Double Check

Ensure that the database should be wiped and that it is ok to lose *everything* in the database.

9.1.2 Step 1: Backup

Backup the PostgreSQL database, ensure that the database dump file has a reasonable file size (dozens of MB to GB, depending on how much test data was crawled, but definitely larger than 0-1 MB).

9.1.3 Step 2: Truncate Database

**Warning:** This operation will wipe report data from the database. Proceed with caution.

Log into pgadmin (or psql), under the “Databases” tree, click on the “montage” database, so that it does not have a red “x” through its icon. In the toolbar, click the “SQL” icon (magnifying glass, which has a hover that says “Execute arbitrary SQL queries”). Enter the following SQL in the top pane, labeled “SQL Editor”:

```
TRUNCATE TABLE montage_core_report RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_core_event RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_core_examtype RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_core_modality RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_core_organization RESTART IDENTITY CASCADE;
```
TRUNCATE TABLE montage_core_provider RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_core_sourcesystem RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_analytics_graphcache RESTART IDENTITY CASCADE;
TRUNCATE TABLE montage_analytics_graphrequestlog RESTART IDENTITY CASCADE;

When ready, click the green arrow in the toolbar (“Execute Query”)

9.1.4 Step 3: Clear Cache

In a cmd.exe terminal, cd to C:\montage\lib\montage-VERSION\bin, and run:

montage.exe clear_cache

9.1.5 Step 4: Recrawl / Re-index

As needed, perform a recrawl and re-run the indexer. It is recommended to restart the “Montage Searchd” service after the re-index.
10.1 System Architecture

Montage is a web-based application, that is hosted within a customer’s intranet. Montage crawls existing data sources to provide fast search and analytics. Users can interact with the Montage search and analytics application via HTTP (or HTTPS) using a standard web-browser (IE6+, Chrome, Firefox, Safari).

10.2 Security Model

Montage uses a role-based access control system. Users must authenticate (either against Montage’s internal authentication system or against an existing LDAP/ActiveDirectory solution).

Once authenticated, user authorization is checked to ensure each user is allowed to access or view the requested part of the application. Authorization is defined within the Montage administrative panel via permissions, which can be assigned either at the user or group level. Each user object can be assigned to multiple groups.

Users can optionally be required to select their “purpose” for using Montage (e.g. clinical, teaching, research, etc.).
Whenever a user views data, the event is logged to an audit log, which resides on the filesystem of the server.

### 10.3 Data Model

Montage primarily crawls report text and associated “meta-data”:

- Accession Number
- Patient demographics (MRN, date of birth, sex, name)
- Who did what and when (e.g. when the report was ordered, completed, signed, etc. and by whom).
- Exam details (stat exam, procedure code)

However, Montage does not, by default, display any PHI meta-data. For example, the MRN and patient name are not displayed to the user (they are only used when linking between multiple source systems). The patient’s date of birth is not displayed and only used to compute the patient’s age at time of exam.

### 10.4 Frequently Asked Questions

**Where does the data reside?** The Montage server resides within the customer’s data center. Indexing occurs completely within this server.

**Does Montage support LDAP / ActiveDirectory?** Yes. Montage can use an internal authentication system or an existing LDAP / ActiveDirectory solution.

**Can the web application be secured with HTTPS / SSL?** Yes. An SSL certificate can be deployed on the Montage Server. Please work with Montage support to obtain and deploy a valid SSL certification for the server.

**Does Montage use Google™ Custom Search?** No. Montage does not use Google’s services or technology. All data remains on the Montage server.

**Does Montage require any browser extensions (e.g. Flash, Java)?** No.

**Does Montage have password requirements (aging, complexity, etc.)** Yes, but must be provided via LDAP/ActiveDirectory.

**Has Montage undergone any privacy or security reviews?** Yes. The privacy offices at multiple sites have reviewed and approved Montage. Montage has undergone an external privacy impact assessment.

Montage has addressed sites’ concern, including adding the ability to require users to identify their “Purpose for Use” and requiring users to accept site-specific Terms of Services.

**Does Montage use a firewall?** Yes. Refer to *Firewall* for specific inbound port requirements (HTTP/HTTPS, optionally SSH/RDP, and MLLP for HL7).

**How is the server secured?** The server only requires inbound HTTP (HTTPS) connections. For management, SSH (on Linux) or RDP (on Windows) are required for approved site system administrators and Montage support.

**Can Montage use a virus checker?** Yes, as provided by the site.

**What occurs during a ODBC “crawl”? What permission is required in the source database?** Montage will connect to the source database, issue a SQL query to find any new reports or reports updated since the last crawl. The crawler then examines those reports and adds it to the Montage index.

The crawler needs read-only permission (the ability to run a *SELECT* statement).

**Is Montage protected against common security vulnerabilities?** Yes, Montage uses tools to prevent common web-based attacks, such as Cross-Site Scripting (XSS), SQL Injection, Cross-Site Request Forcery (CSRF), etc.
Montage must be run with the appropriate firewall enabled.

Ports Montage can use

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>SSH (Linux only)</td>
</tr>
<tr>
<td>80</td>
<td>HTTP</td>
</tr>
<tr>
<td>443</td>
<td>HTTPS (^1)</td>
</tr>
<tr>
<td>3389</td>
<td>Remote Desktop (Windows only)</td>
</tr>
<tr>
<td>6661</td>
<td>HL7 MLLP (^2)</td>
</tr>
</tbody>
</table>

11.1 `ufw` on Ubuntu

To allow SSH, HTTP, & HTTPS

```shell
sudo ufw allow 22
sudo ufw allow 80
sudo ufw allow 443
sudo ufw enable
```

To view the currently enabled ports:

```shell
sudo ufw status numbered
```

To only allow HL7 traffic from a certain host (10.10.10.60 in this case):

```shell
sudo ufw allow from 10.10.10.60 to any port 6661
```

11.2 Windows Firewall

Windows should have Remote Desktop enabled (port 3389) instead of SSH. See *Windows Firewall.*

\(^1\)If SSL/TLS is used, port 80 issues an HTTP 301 Redirect to port 443 to force all traffic over the encrypted channel

\(^2\)By default the HL7 MLLP server is disabled, if disabled the port should be closed
MONTAGE LOG FILES

Logs are typically stored at `/var/log/montage` on Linux and `C:\montage\data\logs` on Windows.

12.1 montage-audit.log

Audit log that lists all reports (by accession number) that a given user potentially saw when performing a search. Also includes all reports exported by a user.

12.2 montage.log

`montage.log` is the primary application log for Montage. It contains general information messages as well as errors that occurred during usage of the Montage application. This log file should often be the first place to check when researching issues.

12.3 montage-importer.log

A log of events and errors during the crawling process.

12.4 montage-importer-hl7.log

Events and errors for the HL7 listener.

12.5 searchd.log

Details concerning the starting & stopping of the searchd service.

12.6 Apache Logs

The Apache logs exist outside the Montage log directory, typically at `/var/log/apache2` on Linux or within the Apache installation directory on Windows (often at `C:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs`).
12.6.1 (montage-)access.log

Generic access log, providing the user’s IP address and the page they requested.

12.6.2 (montage-)error.log

The error typically captures catastrophic errors in Montage, errors that prevent the user from opening the Montage website. The log will often give an traceback that can be used to determine the underlying problem (possibly an incorrect directory in the VirtualHost or a syntax error in settings.py.)
FREQUENTLY ASKED QUESTIONS

Also see Windows Frequently Asked Questions.

- Can Montage be secured with HTTPS / SSL?
- Can Montage be scaled or clustered?
- Can Montage crawl our RIS / LIS / HIS / EMR / etc.?
- Can Montage use a different database or webserver?
- What authentication systems does Montage support?
- How often does Montage crawl / index data?
- Does Montage perform Backups?

13.1 Can Montage be secured with HTTPS / SSL?

Montage can operate over HTTPS. Typically the Apache configuration is moved into the port 443 VirtualHost, which has the SSLEngine set with the certificates. In place of the port 80 VirtualHost, we set a HTTP 301 redirect to the HTTPS version of the server:

```apache
<VirtualHost *:80>
  ServerName montage.example.com
  Redirect 301 / https://montage.example.com/
</VirtualHost>

<VirtualHost *:443>
  SSLEngine On
  SSLCertificateFile C:/ssl/example.com.crt
  SSLCertificateKeyFile C:/ssl/private.key
  # if the chain file is required
  * SSLCertificateChainFile C:/ssl/chain.crt

  # ... standard Montage configuration ...
</VirtualHost>
```

13.2 Can Montage be scaled or clustered?

Yes, the Montage Architecture, is designed to be scalable, but please contact Montage Healthcare Solutions to properly configure.

Often, the easiest method for scaling is simply using a more powerful server.
There are a number of Montage configurations available for scaling up, depending upon the customer’s specific requirements. Specific services can be move to separate servers or Montage can operate in a master-slave mode.

13.3 Can Montage crawl our RIS / LIS / HIS / EMR / etc.?

In most cases, yes! We have successfully integrated into a number of information systems. Please work with Montage Healthcare Solutions to identify if we have already developed a crawler for your information system or how we can go about developing a custom crawler.

Montage’s crawler infrastructure is highly adaptable and can work with a wide range of data formats, including:

- SQL databases
- Non-relational document stores
- HL7 data streams and files
- Comma Separate Value files
- XML files
- Plain text files
- Web services

So far, we have successfully crawled every information system we have been presented with.

13.4 Can Montage use a different database or webserver?

In terms of crawlers, Montage can connect to any database system.

Montage internally uses PostgreSQL and Apache, as part of its “black box” that makes everything run so quickly. At this time, Montage does not support any other database or webserver within its “black box.”

The Montage application is written in a portable manner. It is technically feasible that someday Montage may support other database backends or webservers. However, we make use of some important database features that not all database vendors provide (e.g. strong transactional support, geospatial indexing, etc.). So it would require a fair amount of development effort as well as significant ongoing support and testing effort to use a different database backend.

13.5 What authentication systems does Montage support?

Montage currently supports 2 methods for authentication:

- Internal password storage. Montage will securely (using a salted, one-way hash) store users passwords. The method is secure, simple, and robust, but it does introduce “another password.”
- LDAP / ActiveDirectory. Montage can be configured to defer authentication to LDAP or ActiveDirectory. This solution is ideal if a customer requires specific password requirements (expiration / complexity / etc.).

For both methods, authorization still occurs within Montage, requiring permissions for each account to be configured within Montage.

Please contact Montage Healthcare solutions if you wish to use a different method of authentication, so that we can evaluate the customizations required to plug it into our authentication framework.
13.6 How often does Montage crawl / index data?

As of Montage 2.1, we typically crawl each source system once a day (over night), followed by a complete rebuild of the index files. Index file

In the development version of Montage (not yet released), we have the ability to perform “delta indexing”, in which the new and updated reports are re-indexed when available.

13.7 Does Montage perform Backups?

See Backing up Montage for details on the files & folders that need to be backed up, as well as scripts for creating database exports. Montage does not ship with an automated backup solution, so it is left to the customer to set up scheduled tasks to perform the backups and move the data to a safe location.
SUPPORT

See the *Troubleshooting* guide for common issues.

Please contact support@montagehealthcare.com with any issues or questions. Email is the most efficient means of contacting Montage Support, since it can be routed to the appropriate resource. A toll-free number is also available, (800) 476-0013 x2.

Including any relevant errors from the *Montage Log Files* (making sure to anonymize or remove any PHI) can assist in diagnosis of the issue. Please also indicate what customer is experiencing the issue.
MONTAGE GLOSSARY

Source System  Typically, each “Source System” is a distinct information system. Montage uses the Source System of each report to determine how to format the report (e.g. some information systems store reports with newline characters between lines, while others store the reports with HTML `<br>` characters.

Several examples:

- A site with a single Radiology index and a single PowerScribe 360 installation would have a single source system–PowerScribe 360
- A site with both Radiology and Pathology indexes, coming from two a RIS and a LIS, would have two source systems, the RIS and the LIS.

Please contact Montage Healthcare Solutions to determine the appropriate configuration when a customer has a complex organizational or informational architecture.
This document provides changes relevant to the installation and support of Montage. Please refer to the Release Notes in the User Guide for end-user features and bugfixes.

### 16.1 3.0 Changelog

Montage 3.0 includes numerous new features, a modern user interface update, and numerous usability and performance improvements. Consult the Montage User Guide for complete, user-facing, 3.0 release notes.

In terms of the server installation and management, the release includes:

- Faster upgrades, doesn’t reload data that already exists
- Updated to latest RVU file
- Improved HL7 support (exam modifier, technologist, additional timestamps)
- Improved speed for IDX crawler
- New Cerner crawler
- Chrome Frame is no longer supported by Google, so we have removed the suggestion to use it
- Data Integrity Checks
- The Tag Admin now features a fast, easy to use interface to create and manage Tags
- The Montage Admin has the capability to disable specific QC Algorithms

### 16.1.1 Deprecated Features

- Montage 3.0 has dropped support for Internet Explorer 6.0 and 7.0. Internet Explorer 8.0 is supported, but lacks native HTML5 support. Montage works around these issues, but an improved user experience is provided by modern browsers such as Internet Explorer 10, Firefox, Chrome, and Safari.

### 16.1.2 Upgrading

A database backup is highly recommended prior to upgrade.

If upgrading from 2.5, run the standard upgrade process. On Windows, stop the Apache and Montage services, run the installer.

If upgrading from a release prior to 2.5, you can directly upgrade to 3.0, but be sure to read the notes and extra steps in the intermediate Changelog entries, e.g. the 2.5 Upgrade.
16.2 2.5 Changelog

**Warning:** The upgrade to 2.5 includes several intensive, automated maintenance tasks. See the upgrade instructions for details before starting the upgrade.

Montage 2.5 includes numerous new features, as well as usability and performance improvements. Consult the Montage User Guide for complete, user-facing, 2.5 release notes.

In terms of the server installation and management, the release includes:

- New optional features such as *Residency Dashboard* and *Group Organizations*
- A powerful new *PACS launch system*
- Several *data model additions* and *crawler updates*, such as Point of Care support
- *Performance improvements*
- *Documentation* updates
- Numerous other *Additional Changes & Improvements*

---

16.2.1 Residency Dashboard

Montage 2.5 includes a new Residency Dashboard under Montage Analytics for tracking residents progress towards the ACGME requirements.

This feature is not enabled by default, and must be enabled in `settings.py` with `MONTAGE_RESIDENCY_ANALYTICS`.

Additionally a local administrator or residency program manager must configure set up the residents’ expected graduation date in Montage Admin via `Admin > Providers > Edit`.

16.2.2 Group Organizations

If a site wishes to restrict which Organizations individual users or groups have access to, there is a new setting, *Group Organizations*. 
It is recommend only to enable this setting if absolutely necessary, since it creates additional complexity in user management (groups must be granted explicit access to sites). Much of the power of Montage comes from having all the data available.

### 16.2.3 Improved External Launcher

The ability to launch PACS has been revamped, see *External Launcher* for details.

A new permission exists to restricting access the the launchers, “Can use External Launchers (e.g. PACS)” permission. Audit logging can optionally be added now with the *Redirect Launcher* which submits to PACS launch request back to the Montage server before sending it to the PACS.

Contact *Montage Support* for assistance configuring.

The prior `Index.external_launcher()` approach is now *deprecated* and will be removed in an upcoming release.

### 16.2.4 Crawlers

- Ability to support sites where Exam Codes are not unique via `exam_codes_are_unique`
- Ability to support accession number collisions (requires configuration by *Montage Support*).
- Point of Care available for PowerScribe 360, IDX, and HL7. Enabled by default.

#### PowerScribe 360

- Ability to ignore reports with specific exam codes from ever being crawled via `excluded_exam_codes`
- *Compatibility chart* for Nuance crawler support
- Crawling additional data, including Point of Care, Patient Middle Name, & Ordering Provider NPI

#### IDXRad

- Force conversion of IDX Provider IDs into integers to avoid differences in SQL Server ODBC drivers

#### Syngo

- Improve the report content concatenation when dealing with backloaded reports
- Expand the crawler’s query filters to capture very late signs & addendums

#### ClearCanvas

- Ideally crawl from ReportText, but fall back to ReportContent, as used in older versions of ClearCanvas

#### QuadRIS

- QuadRIS crawler included
HL7

A major update occurred in Montage’s HL7 processing. For the most part, this will not affect any sites. However, if the site is currently using a custom HL7 profile from the config folder, please contact Montage Support to switch to an updated, built-in version of the profile.

16.2.5 Data Model

• Point of Care
• Additional data elements added to the Montage schema for future usage
• Configurable patient matching

16.2.6 Performance Improvements

• RAM usage drastically reduced when exporting large datasets to CSV and XLS. It is possible, if desired to increase the 10k limit via MONTAGE_MAX_EXPORT.
• Structured the schema and text layout which should result in much faster performance in Montage Analytics. Because of the large changes and automated maintenance, the upgrade process may take longer than normal. See the upgrade instructions for details
• Optimized several queries for faster lookups in the Patient Record and under certain conditions in Montage Analytics.

16.2.7 Additional Changes & Improvements

• Allow building of spell suggestions on Windows, via montage.exe build_spell_suggestions C:\montage\config\searchd.conf May take several hours to build.
• Automatic upgrade of mod_wsgi on Windows
• Monitor system performance (server stats, analytics & crawler performance) via the stats upload
• Allow site-customization of the ordering of the Exam Type
• Remove ASCII SUB character from CMS RVU import
• Removed montage change_password backport for Windows, the original changepassword is preferred when having to forcefully reset a user’s password from the command line. It is still recommended to use the Montage Admin’s “Reset Password”.

16.2.8 Documentation

The Settings documentation has been added to and split into individual pages related to groups of settings. Important new sections include:

• External Launcher
• Residency Analytics
• Increasing the export limit

A new Compatibility chart exists for Nuance systems.
16.2.9 Deprecated Features

- `Index.external_launcher` is deprecated in favor of the new `External Launcher` setting. Existing PACS launches should be migrated to the new system prior to Montage 3.0.

16.2.10 Upgrading

The upgrade to Montage 2.5 requires an automated, one-time schema change and maintenance operation (for the `Performance Improvements`) that will take longer than prior upgrades, requiring some downtime of the Montage system.

Before starting the upgrade, ensure that the PostgreSQL data drive or partition has sufficient free space (e.g. not 90% full).

A `database backup` should be taken prior to starting the upgrade.

Ensure the customer is aware of the downtime and schedule according. The process may take 20 minutes to two or more hours for large installs.

Run the standard upgrade process. On Windows, stop the Apache and Montage services, run the installer. The Command Prompt window will stay open while the migration and maintenance occurs. Do **not** kill the upgrade process once started, it may take significant time. Contact Montage Support if you are concerned it is stalled (the install.log file in the installation directory contains additional details). Once complete, start the services as normal.

Run Windows Disk Defragmentation on any disks that have high fragmentation levels to complete the maintenance tasks.

16.3 2.4 Changelog

Montage 2.4 includes additional QC algorithms and additional analytics capabilities. Consult the Montage User Guide for complete, user-facing 2.4 release notes.

In terms of the server installation and management, the release includes:

- Several crawler additions and modifications
- Updates to documentation
- Additional bugfixes and updates
- Performance improvements

- Crawlers
  - Nuance PowerScribe 360
  - GE IDX
  - ClearCanvas
  - Others
  - Geoanalytics
  - HL7
- Server Updates
- Documentation Updates
- Upgrading
16.3.1 Crawlers

**Nuance PowerScribe 360**

- Better support for report addendums. If a customer has experienced missing addendums, historical data can be recrawled (contact Montage Support to discuss).
- Montage can be limited to only crawl specific Site instances from PS360, instead of crawling data from all Sites. For details, on configuring, see `limited_site_names`.
- Crawling of patient zip for Geoanalytics.
- Ability to configure how to handle addended reports that were imported and not authored in PS360 via `overwrite_on_imported_addendum`.

**GE IDX**

- Montage can be limited to only crawl specific Organization’s data, via the `limited_organization_ids` crawler param.
- Better support for Residents and additional timestamps. instead of crawling data from all Sites.
- Crawling of patient zip for Geoanalytics.

**ClearCanvas**

- Do not crawl & actively remove patients with Withdrawn Consent.

**Others**

- Additional site-specific crawlers.

**Geoanalytics**

Montage 2.4 features some basic geo-analytics, for some (not all crawlers), Montage will now try to pull the patient’s address, including city, state, and Zip. This new functionality can be disabled if required by the customer via the `ignore_patient_geo` crawler setting.

**HL7**

- Patient Status is now pulled by default from PV1-2.
- Is Stat is now pulled from ORC-7.6.
- Preliminary and Addendum messages handled via OBR-25.
- Ordering timestamp can optionally be set in the ORU^R01 message now, but defaults to off.
- Modality can optionally be pulled from the message, but is not by default.
16.3.2 Server Updates

- The Windows installer no longer prompts for an install directory (since it had to be C:\montage). The installer proceeds and automatically installs into C:\montage.

- Prior to Montage 2.4, several Python packages (lxml, psycopg, etc) had to be installed manually. As of Montage 2.4, these are installed automatically. If previously installed, you can safely remove these libraries via the Control Panel, though it is not required.

- The latest RVU-CPT mapping is automatically loaded (no longer requires Montage Support to load). Alternate RVU mappings can be loaded via import_rvu. Additionally, Montage has updated to that latest available RVU values.

- HTTP Proxy support when outbound Internet access must go through a web proxy to upload health stats and query Yottalook. See MONTAGE_REQUESTS_PROXIES.

- On Windows, we have replaced our file-based cache with an in-process memory cache system, which should dramatically increase performance for concurrent requests to Windows-based Montage installations.

**Warning:** C:\montage\config\settings.py should no longer use the CACHES[‘default’][‘LOCATION’] = ‘D:/montage/data/cache’ directive to move the Cache to another drive, since the cache is now in memory. This setting line **must** be removed as part of the 2.4 upgrade.

- Custom logo on the login page is now supported on Windows Server. Can be uploaded via the Admin > Customer Logo page.

- Support HTTP Basic Authentication on Windows.

16.3.3 Documentation Updates

Additional documentation has been added as of Montage 2.4, including:

- Pre-install PowerScribe 360 Data Analysis Script, to be run to identify common data mapping issues prior to executing the crawl.

- Database Truncation for wiping the database. Useful if Montage has a crawled a Test PS360 instance

- Added note that Microsoft IIS must not be installed.

- Server Specs have been updated with new requirements and additional clarification.

16.3.4 Upgrading

Please follow standard upgrade procedures (stop services, run installer, restart services). If data needs to be re-crawled to obtain patient zip data or missing addendums, please do so.

16.4 2.3 Changelog

Consult the Montage User Guide for complete, user-facing 2.3 release notes.

Montage 2.3 includes major performance improvements as well as numerous user-facing features. Several crawler adjustments have occurred that may require adjustments to enable new features (e.g. Montage QC) or to capture additional data (e.g. Residents from PS360).
16.4.1 Crawler Modifications

Potentially Requiring a Recrawl

Several changes occurred that may require a data recrawl depending on if the site would benefit from the additional crawled data.

Nuance PowerScribe 360 / RadWhere

- Residents are now crawled from the Dictator field

GE IDX / Imagecast / Philips iSite 3.6

- Additional timestamps (Patient Arrived, Ordered, Exam Started). The Ordered timestamp will allow for proper operation of the Ordering Provider dashboard
- Residents from vusrContributingProvider

If the site would benefit from this data, a full recrawl is suggested (Initial Crawl, either via --full-import or broken into ranges with --start-date and --end-date). Montage Support can assist in planning and performing a recrawl if necessary

Enabling QC

To use Montage QC, a valid montage_qc entry must exist in montage-modules.json (contact Montage Support for an updated license if not present) and the QC detection must be enabled in the appropriate crawlers.

Within the appropriate params entry of the CELERYBEAT_SCHEDULE crawler, detect_qc must be set to True. Once Montage CeleryD and Celerybeat are restarted, the scheduled crawl will automatically start checking every report for all the configured algorithms.

The new --detect-qc option is available for the run_importer command, but is not recommended for full-imports, little value exists for for old QC events and the QC engine will dramatically increase the CPU requirements during a crawl.

Note: Montage QC is still a beta feature. If there are mis-identifications, Montage Support is interested, so the precision and recall can be improved. Montage QC uses complex natural language processing of inherently imprecise human language, so the system will always have some mis-identifications. Also note that many additional algorithms are developed and will soon be published.
Enabling Radimetrics eXposure

It is recommended to work with Montage Support to enable the Radimetrics feature, since Montage Support must work with Radimetrics Support to ensure the correct components are present on the eXposure server.

- The montage-modules.json must have a valid entry for montage_radimetrics
- The settings.py must have the Radimetrics settings configured
- The crawler configuration should have crawl_radimetrics enabled
- The Montage Searchd configuration file must be regenerated and the indexer run to re-index
- The Apache, Celeryd, and Celerybeat services must be restarted

The new --crawl-radimetrics option is available for the run_importer command, but is not recommended for full-imports, since a webservice call is made for each report.

16.4.2 Other Backend Changes and Notes

- Montage Support has noticed several Windows servers having high disk fragmentation. If a server is running slowly or has high fragmentation, Montage Support suggests working with the customer to turn on scheduled defrags using the Windows Disk Defragmenter.
- Can disable crawling Modalities with import_modality crawler setting. This option requires manual, backend management of Exam to Modality mapping. Future releases will include an admin-focused tool for managing this mapping.

New documentation and guides:

- New pre-install and post-install checklists to assist during the requirements gathering, implementation, and checkout phases. Also, additional Nuance-specific pre and post install notes.
- Backup documentation and an example PostgreSQL backup script in Backing up Montage

- Expanded Settings to include Email, LDAP, Authentication, and Purpose of Use
- Expanded the Nuance with additional options

Also, Montage will now automatically perform a weekly database maintenance task (VACUUM ANALYZE within PostgreSQL) to improve database performance by removing outdate pages.

16.4.3 PostgreSQL 9.1.9 on Windows

Due to CVE-2013-1899, existing Montage installations on Windows, must upgrade their PostgreSQL installations from 9.1.1 to 9.1.9. The CVE is low risk, since the Montage implementation requires PostgreSQL access to be blocked at the firewall.

It is suggested, though not required, to backup the PostgreSQL database before proceeding. Since it is a minor point release, the PostgreSQL upgrade will upgrade the existing installation in place.

The new Montage installation package includes the postgresql-9.1.9-1-windows.exe installer. This can be run during the upgrade process below, after the services have been stopped, before the file: montage-2.3.0.exe installer is run.
16.4.4 Upgrading from 2.2

Ubuntu

```bash
sudo dpkg -i montage_2.3.0-0_amd64.deb
```

Redhat

```bash
sudo rpm -Uvh montage-2.3.0-0.el6.x86_64.rpm
```

Windows

Obtain the montage-install-package-2.3.0final-0.zip package. Only the montage-2.3.0.exe and postgresql-9.1.9-1-windows.exe are required for upgrades.

If any settings adjustments are required, perform them (e.g. enable Montage QC, etc.).

2. Upgrade to PostgreSQL 9.1.9 on Windows. The PostgreSQL installer will automatically pick up the existing database location.
3. Run the installer, montage-2.3.0.exe.
4. Start the Windows Services (Montage Celeryd, Montage Celerybeat, Apache 2.2). Only if used at the site, Montage MLLP Server.

16.5 2.2 Changelog

16.5.1 Upgrading from 2.1

The standard upgrade process can be used between 2.1 and 2.2. No special steps are required.

The automatic database migration will take more time than usual as all prior Turn around times will be pre-computed and stored. The time for migration depends on the number of reports and the speed of the server. At most sites, it takes between 5-30 minutes.

16.6 2.1 Changelog

16.6.1 Upgrading from 2.1.0-beta3 to 2.1.0-final

- Montage MLLP, Montage CeleryD, and Montage CeleryBeat changed from Scheduled Tasks to Windows Services
- pywin32 dependency added as part of Python Libraries installation
- Web-based post-install configuration tool.
16.7 2.0 Changelog

Upgrading from 1.0 to 2.0 requires a fresh install, as it encompassed a major re-write. Contact Montage Support for assistance.
INDEX

S
Source System, 141