Application Depot

Preface

Application Depot (AppDepot) is a web application and user management system. Its raison d'être is to provide a secure platform where users are matched with web applications. While it is true that there are many content management systems (CMS) and portals in existence today with this capability, AppDepot is neither. At its very essence AppDepot is a shell, much like a bank vault with safety deposit boxes; whose purpose is to provide security while allowing users unfettered access to authorized objects. AppDepot has secure portal features but goes further in providing utility and management functionality.

AppDepot is not a content management system so web applications are not stored in AppDepot, nor written in any AppDepot specific language. It can use existing web applications (remote or local). AppDepot exchanges credential information with the application prior to the start in order to authorize the use. The exchange of credentials allows the application to determine if it should run or not. This is done via a small and simple prologue script inserted into the web application or via the AppDepot API. Alternatively AppDepot can also execute web applications without the security wrapper; which mimics regular portal functionality.

Additional to the application authentication is the user management capability. AppDepot supports multiple LDAP sources for increased flexibility. LDAP authentication relieves AppDepot from managing people and passwords. AppDepot uses the LDAP source to identify and authorize user accounts, and map authorized web applications to the user.

Web developers can speed up their development with the use of AppDepot by relieving the programmer of security concerns; the developer is free to focus on core functionality and role programming. Core functionality are features which define the application and give purpose to the user. Role programming is the idea that a web application can be segregated into distinct modules and made available to only those users filling a certain role. For example, data custodians might have access to the module responsible for maintaining the applications data, while administrative assistants might only need the reporting component for their job. Web application modules can then be written specifically for each role, and modified and tested independently.

AppDepot is lightweight and designed to operate with a low profile. Web applications remain the focus of the user's attention while AppDepot manages the backend.

About

Application Depot (AppDepot) was written to simplify web development while providing security. In today's Internet environment security is paramount, unfortunately, securing applications is time consuming and expensive. User accounts have to be managed and data access has to be restricted. This problem is compounded by the addition of more web applications. Over time user accounts fall out of sync across applications and support and maintenance difficulties increase.

To be sure, there are content management systems (CMS) available both in the public domain and the commercial world. These CMS systems solve the problem of user authentication and data authorization. They are feature rich and promise flexibility and simplicity. But many require that the development team learn a new API or some esoteric management system. And despite their (CMS) best intentions, they end up hemming the team by restricting the development method, eventually limiting the team's capabilities. This process often leads to a search for a new CMS system requiring a migration of legacy applications; and the process begins anew.

The developers of AppDepot took a different approach towards secure web application development. Instead of trying to be everything to everybody why not focus on critical time consuming issues? – Web security and account management. This frees up web application developers to spend more time on the application's core functionality, knowing they can securely wrap their application with AppDepot when they're ready to deploy.

Specifically, AppDepot attempts to adhere to these security principles:

- *"Economy of mechanism": Keep the design as simple and small as possible*
- *Fail-safe defaults: Base access decision on permission rather than exclusion.*
- *Open design: The design should not be secret, but rather depend on the possession of keys or passwords.*
- *Least privilege: Every program and every user of the system should operate using the least set of privileges necessary to complete the job.*
- *Psychological acceptability: It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly."* (Information Security)

AppDepot provides four basic functions for the development team: application security, account management, user and application mapping, and modular application development capability (role development).

Application Security

AppDepot is designed as a security wrapper for web applications. The application is registered with AppDepot and a unique application ID is generated.   This unique ID is then used in the application's prologue. Before the application starts AppDepot and the application authenticate each other. It is up to the web application to determine whether the 'hand shake' is valid before it executes. This philosophy places the onus on the web developer to ensure the requesting call is legitimate. The prologue is very small and simple to implement.
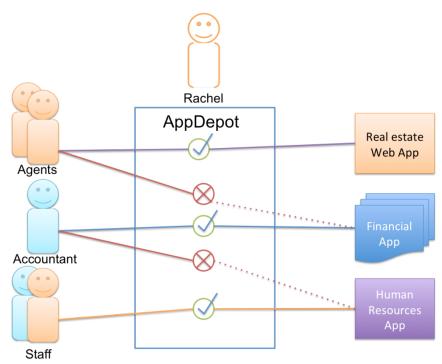
User and Application Mapping

Figure 1 User and Application Mapping

AppDepot supports account login and application mapping. Before a user can log into AppDepot the administrator must add them to their instance of AppDepot. The user must have a valid account and password in an operational LDAP server. When a user has been added the administrator will map the applications they will be allowed to access. In this scenario, Rachel, owner of a real estate business and the AppDepot admin, has three applications with differing functionality. She makes specific web applications available only to those individuals needing access.   The account and application management tools are simple to understand and use.
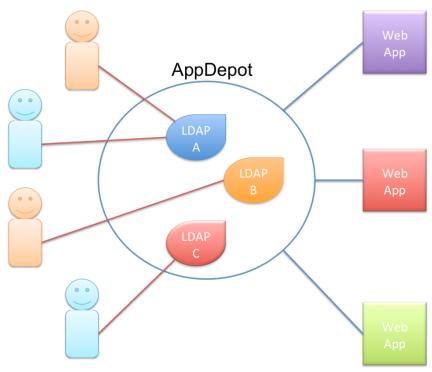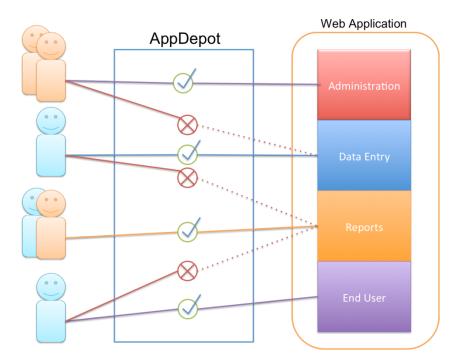
Account Management

Figure 2 LDAP Support

AppDepot supports multiple LDAP authentication sources for those heterogeneous environments where clear divisions reside. This capability allows users from disparate domains to securely access commonly used applications. The AppDepot administrator does not have to deal with the account management aspect only the authorizing source. AppDepot supports most LDAP implementations from Microsoft Active Directory to Sun's Directory Server.

Modular Application Development

The controlled environment AppDepot creates makes it possible for web application developers to fully modularize their applications. Instead of developing a wholly contained application with administrative, reporting, querying and maintenance capabilities, an application can be separated into its constituent parts. Each component can be developed independently and deployed inside AppDepot. This reduces the difficulties with modification, testing and maintenance of the entire application. The components are tailored for the user (role), simplifying and expediting the development process. By employing this methodology an application will evolve into a series of smaller applications each suited for a particular need, yet loosely tied together as the complete application.

AppDepot is not a content management system. The main objective of AppDepot is to offload critical functionality from the application and unto AppDepot. Web application developers are then free to focus on core functionality.   AppDepot works to provide a lightweight, barely visible platform where users and applications meet securely.

Application development and Integration

Integrating Applications into AppDepot can be done on many different levels using either the ADBouncer system (javascript or PHP) or the Application Depot API. The ADBouncer system is essentially the necessary API calls rolled into a simple 'drop in'
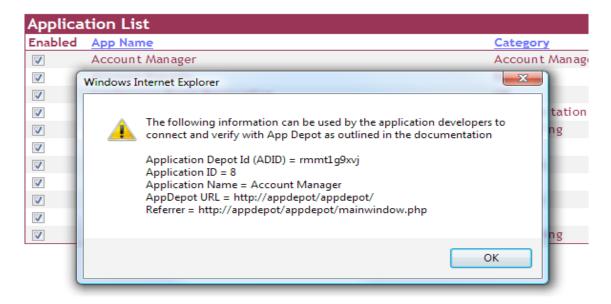
program for ease of use. The API itself, however, is available for application developers themselves to use if so desired.

Both methods also return variables back to the application that can be used by the application developers for many different purposes:

- adusername: The App Depot user name of the user logged in
- aduserlevel: The user level of the user logged in (5=user, 0=admin)
- adfullname: The full name of the user logged in
- aduseremail: The email address of the user logged in
- adauthsource: The LDAP authentication source used
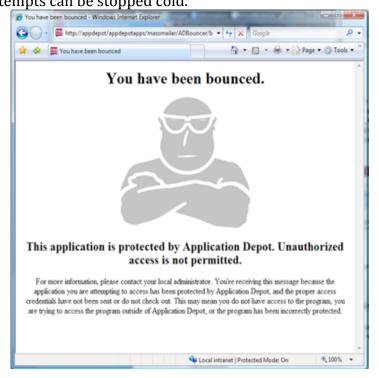
External Application Credentials

External Application Credentials (EAC) are the variables used to provide applications the information they need to establish a successful handshake with AppDepot. Each application will have its own set of EAC and these credentials will be available to any AppDepot administrator via the Administration area, Application Administration. Click on the EAC link next to an application to get the EAC for that app.

EAC are used by the ADBouncer system to verify that an application has been properly secured and is being accessed by a valid user with a valid session.

ADBouncer

The ADBouncer is a set of prewritten 'drop-in' programs that make it easier for developers to integrate AppDepot authentication into their applications. There's no way for AppDepot to 'know' that an external application is being accessed insecurely so it cannot stop such access on its own. Coupled with the ADBouncer, however, such access attempts can be stopped cold.

ADBouncer can be integrated into your applications as follows:

1. Insert a copy of the ADBouncer folder found in the AppDepot archive into the root directory of the application you wish to protect.
2. You may use the PHP version of ADBouncer or the Javascript version.
    2.1. Javascript Version: *bouncer.js*
        2.1.1. Works across all languages
        2.1.2. Less secure than server side methods
    2.2. PHP Version: *bouncer.php*
        2.2.1. Works with PHP applications or web pages
        2.2.2. Server Side Secure
    2.3. Other Versions
        2.3.1. No other versions currently exist... write one!
3. Choose the version you like, and edit the EAC information at the top of the file to match the EAC information given by AppDepot
4. Choose your security level
    4.1. **0**: No security checks are done
    4.2. **1**: Checks that the App Depot ID is available and valid
    4.3. **2**: Additionally Checks that the referrer is as expected

4.4. **3**: Additionally performs a handshake with AppDepot itself and verifies that the logged in user has access to this program and that the session is valid.
Note: A security level of 3 requires that the external application is capable of communication with your AppDepot installation and is not blocked by any firewalls or other security.

5. Drop in the necessary include code in every page you wish to protect.
    5.1. JS ex: *<script type="text/javascript src="ADBouncer/bouncer.js"></script>*
    5.2. PHP ex: *<?php require_once('ADBouncer/bouncer.php'); ?>*
6. If you wish to use any of the return variables, they'll be available to you after the includes have been made. Refer to the variable list at the beginning of this section. Note that they'll be in the format of the ADBouncer version you chose, i.e. PHP variables will be $adusername.

AppDepot API

If you or your developers wish to write your own Bouncer programs, or need a bouncer for a language we haven't already written one for, the AppDepot API is available. Simply the API consists of some variables sent from AppDepot during an application request and a set of scripts in the AppDepot application that can be called and verify the information is correct.

Variables sent when an application request is made. (Sent via GET in url)
*   adid: AppDepot ID. This is an ID number that is unique to each AppDepot installation.
*   sessionid: The ID of the current session in use by the user logged in

Verifying the ADID is the first step in security, the ADID is available via the EAC link in the Application Administration tab in the Admin Section of AppDepot. Verifying that the sent ADID and the ADID provided by the EAC match will prevent the majority of external access violations.

*   PHP Example:
    $eac['adid'] = 'adid given by EAC link';
    if($_GET['adid'] != $eac['adid']) die('unauthorized');

The next step is communicating with AppDepot to verify the session. AppDepot has the script appverify.php located in the AppDepot root directory which can be accessed via an HTTP request to check the session information and verify it is correct. The script returns XML containing the response. appverify.php expects the following parameters to be passed in the url:

*   sessionid: The session ID to verify, sent by the application request
*   appid: The application id provided in the EAC for this application

The following XML is returned on success:

```
<response>
<result>success</result>
<content>Session Remotely Verfied (jdoe,ApplicationName)</content>
<sessionid>avdcejicnb7ntnpkb1cuccsmk6</sessionid>
<username>jdoe</username>
<userlevel>0</userlevel>
<fullname>John Doe</fullname>
<email>jdoe@yourcompany.com</email>
<authsource>Your LDAP Server</authsource>
</response>
```

This XML is returned on failure:

```
<response>
<result>error</result>
<content>Error Message</content>
</response>
```

What you do with either response is completely up to you or your developers. It is recommended that a failure be treated as an unauthorized access attempt.

Log Writing API

AppDepot also allows external applications to write log entries in the AppDepot logs. This may be useful to keep all of your application errors in one place or to have AppDepot report level 3 errors on external programs to the AppDepot administrators. Writing logs is very similar to verifying a session. The log script is writelog.php in the AppDepot root directory. It expects the following parameters to be passed in the url:

- application: The name of the application writing the log
- entry: The **URL Encrypted** log entry
- (optional – default 0) severity: How serious of a log entry is this
    - 0: Routine
    - 1: Mild error or notification
    - 2: Serious error or notification
    - 3: Critical error or notification

Note that level three errors will be listed in the Admin Error Display if you have that option enabled in the AppDepot settings.

The following XML is returned on success:

<response>
<result>success</result>
<content>This will contain the successfully written log entry</content>
</response>

This XML is returned on failure:

<response>
<result>error</result>
<content>Error Message</content>
</response>

Setup and Installation

AppDepot can be downloaded from our website at http://appdepot.org or
http://sourceforge.net/projects/appdepot. Once downloaded, installing AppDepot
is a straightforward process. Simply unzip or untar the files into a web directory on
your server and browse to the root application folder to begin the installation
process. Prior to installation, you'll need to have the following requirements met:

- The PHP user on your server must have read/write permissions to the
  following files:
    - The entire *install* directory and all files inside
    - The *includes/settings.php* file
    - Whatever location you'll specify as the log directory

- You must create a MySQL database and a user for App Depot to use
    - Be sure to give the user permissions to the database you create
    - App Depot will create all tables and installation data for you

- You'll need at least one valid LDAP Authentication Source
    - This can be Active Directory, Open LDAP, or any other authentication
      source that uses the LDAP protocol
    - All App Depot users MUST have an account in a valid LDAP source in
      order to be added to App Depot
    - You'll need to specify your first LDAP source during installation

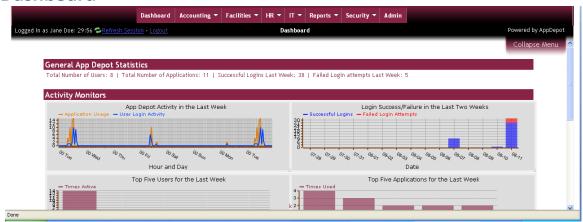- The PHP directive 'register_globals' MUST be turned **OFF**

The AppDepot installation program will verify that all these conditions are met
prior to proceeding with installation. If any of the above conditions are not met, you
will not be allowed to continue.

Follow the instructions on the installation page to continue and complete the installation. At the bottom of the installation page is the 'Verify Information' button. The installation will not complete until all the information you have given can be verified as accurate and all the appropriate connections can be made. Once verification is complete click 'Complete Installation' to finish the process.

Note: Changing any information after a successful verification will require all information to be re-verified.

The admin email is who your users can contact if they have issues. The root directory is the base directory for your App Depot installation. For example, if your site is www.example.com and you have AppDepot installed at www.example.com/programs/AppDepot then your root directory is /programs /AppDepot. The logfile will be used to write logs in case of a database failure.

Admin Email: `jdoe@yourdomain.com`
Root Directory: `/appdepot`
LogFile: `errorlog.txt`

Database Settings

The database settings define how to connect to the database you have setup. If you're running AppDepot on the same machine as your MySQL installation, then host should be 'localhost'. If not, then you'll need to specify the server address or IP.

Database Host: `10.10.0.42`
Database Name: `appdepot`
Database User: `appdepotuser`
Password: `••••••••`

LDAP Settings

You'll need to specify a working LDAP source so App Depot can setup the initial admin user. You'll be able to add additional LDAP sources once App Depot installation is complete.

Source Name: `Your LDAP Server`
Host Address: `ldap://10.10.120.20`
Port: `636`
Connection Username: `CN=John Doe, CN=Users, DC=YourDomain, DC=local`
Ex: *UID=Username,OU=Users,O=company* or *CN=Username,CN=Users,DC=mycompany,DC=com*
Connection Password: `••••••••`
Search DN: `DC=YourDomain, DC=local`
Ex: *OU=Users,O=company,O=com* or *CN=Users,DC=mycompany,DC=com*
Username Field: `samaccountname`
Display Name Field: `displayname`
Email Field: `mail`

Admin User Settings

These settings create the initial admin user for App Depot so you can log in and set up the rest of your users and programs once the installation is complete. Please enter a valid username from the LDAP source you provided above.

Admin Username: `jdoe`

You have made changes to your information, so you must re-verify before continuing

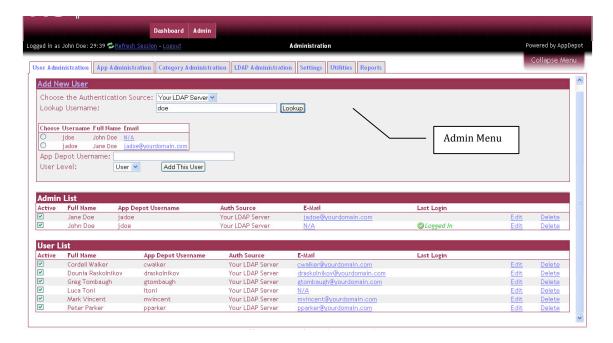[Verify Information]   [Complete Installation]

Dashboard

The dashboard is visible to admin accounts only. It provides a summary of the user and application activity. More detailed reports are available on the via the 'admin' tab.

Administration

User

Adding and editing users in AppDepot is very straightforward through the Admin module. As an administrator, you have access to this module by clicking on the Admin button. Clicking on it will bring up the main Administration page which has several tabs: 'User Administration', 'App Administration', 'Category Administration', 'LDAP Administration', 'Settings', 'Utilities' and 'Reports.

The 'User Administration' tab has 'Add New User' link. An 'Admin List' and a 'User List' is displayed showing the admins and users who have accounts in AppDepot. Active accounts are checked as being Active.

To add a new user, click the 'Add New User' link at the top of the page. This will allow you to do a Username Lookup for a user who exists in your LDAP system. If the user does not have an account or the username you're looking up is not found, an error message will appear next to the user name.

To delete a user, go back on the 'User Administration' tab. Click the Delete link next to the desired username:
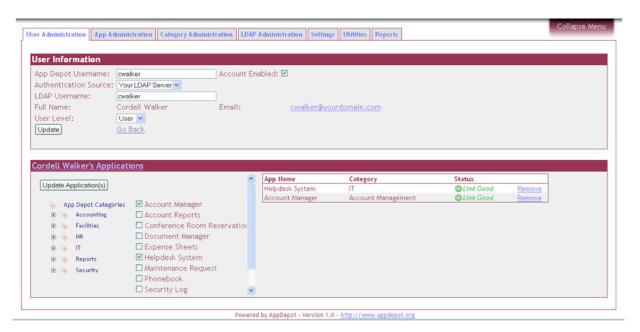

To view 'User Information', click the Edit link next to the desired username. On this page, you may change user levels (user/admin), update user information and enable or disable the user's account by checking the Account Enabled checkbox. Disabling a user's account prevents them from being able to log into the system without removing any of their application permissions or deleting them outright.

Note: Admin users cannot be disabled. They must be 'demoted' to regular users first.

Once a user is added, you can grant access to the applications by clicking on the edit link next to a username. The 'User Information' screen appears. Click on the 'Username's Applications' to display a list of all programs currently integrated into AppDepot. To grant the user access to applications, select or deselect the checkboxes next to the application name and click the Update Application(s) button. You may also click on the Remove link on the right hand pane to remove access to an application.

Note: You may also grant access to applications thru the App Administration tab.

Application

You may add a new application through the App Administration tab and 'Add New Application' link:

You may view, edit or delete applications through the App Administration Tab. Under 'Application List', click on the Edit link next to the application (i.e., Account Manager) that you wish to view or modify.

Click 'Users with access to Account Manager' to view a list of users who have access to the application. The screen shot below shows all the users that have access to the Account Manager application. Select the checkboxes next to the user name and clicking the Update User(s) button to grant the user access to this application.

Category

Click on the Category Administration tab to access the application category menu. Click 'Add New Category' at the top of the Category Administration page. This will display an input form allowing you to enter information about the Category such as Category name, Category Description, Category Parent, etc.

The drop-down menu on Category Parent allows you to choose where your application should reside.
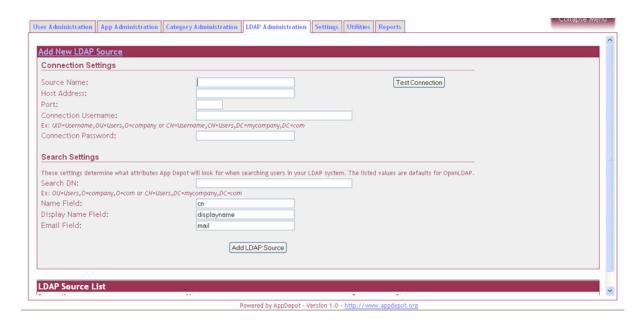

Click on the 'Add Category' button. A visual representation of the Category Structure is shown on the bottom pane.

The expandable category structure allows you to Edit and Delete selected categories as well.

LDAP

App Depot uses LDAP to authenticate all of its users, and you can use many different LDAP sources of different types to bring all of your users together. Before a user can be added, his or her LDAP source must be defined. This is done via the LDAP Administration tab.

Click on the LDAP Administration tab and 'Add a New LDAP Source.' Enter the required information for your source.

- Source Name: This is what you call your source, it can be anything you like
- Host Address: The server name or address on your network where the LDAP source resides. Ex: *ldap://10.10.20.120* or *ldap://auth.mycompany.com*
- Port: The port your LDAP sources uses to connect
- Connection Username: The user from your source that has permission to search the directory, either in user@yourcompany.com format or the full distinguished name
- Connection Password: The password for the above user

Once you have all the correct information, click 'Test Connection' to verify that App Depot can connect and search your directory. Once you have a green light the source has been verified.

The Search Settings portion defines how App Depot will look for new users. If any of these settings are incorrect, the connection to the server will work just fine, but searching for users may return an error.

- Search DN: This is the portion of your LDAP directory in which to search for users. It will differ for each installation, but typically is in the form of CN=Users,DC=mycompany,DC=com
- Name Field: Where your LDAP system stores the username. For Active Directory systems, usually it's samaccountname. In Open LDAP it's usually cn. Your system may vary.

- Display Name Field: Where your LDAP system stores the user's full name.
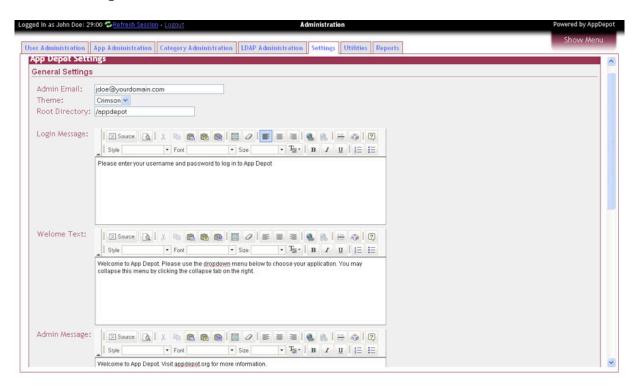- Email Field: Where your LDAP system stores the user's email address.

When you're done click 'Add LDAP Source' to add it into App Depot.

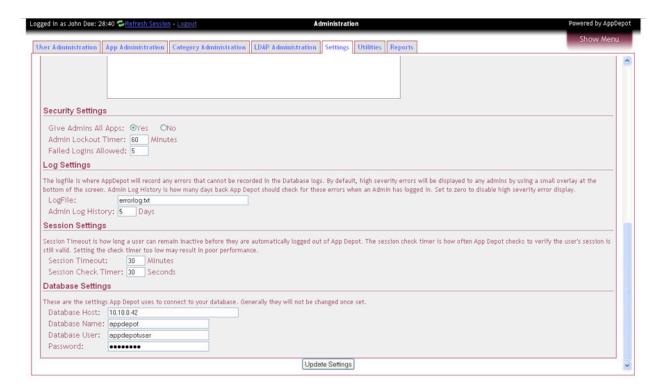You may also view, edit or delete sources from the 'LDAP Source List'

LDAP Source List					
Source Name	Host	Port	Status		
Your LDAP Server	ldap://10.10.120.20	636	Connection OK	Edit	Delete

Settings

General Settings

- Admin Email: The administrator's email address
- The Login Message: The message users see at the login screen
- Welcome Text: The message displayed at the top of the App Depot menu
- Admin Message: The message displayed to users on their dashboard at login

Security Settings

- Give Admins All Apps: Toggles if all admins should have access to all applications by default
- Admin Lockout Timer: How long admins are locked out of they exceed the failed login limit
- Failed Logins Allowed: How many incorrect password attempts are permitted before a lockout occurs
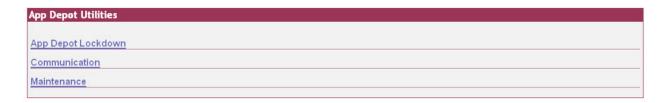
Log Settings

- LogFile: The location and name of the textual log file, used only in the event of a database error
- Admin Log History: Controls how many days back the high severity error display will check for errors. Set to zero to disable the high severity error display

Session Settings

- Session Timeout: How long, in minutes, an App Depot session may remain idle before automatic logout
- Session Check Timer: How often, in seconds, App Depot verifies the user's session. Setting this number too low may result in serious performance issues.

Database Settings are what AppDepot uses to connect to your database and generally won't be changed after the initial setup.
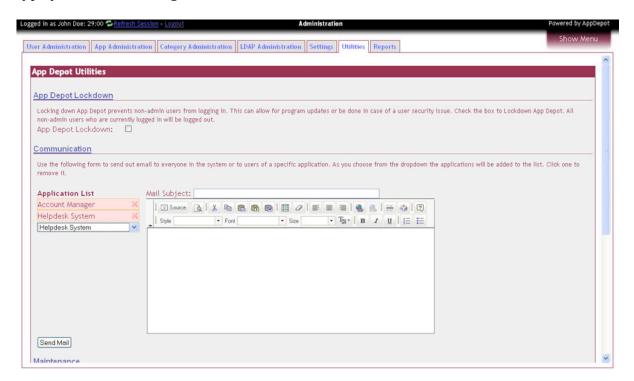
Utilities

The 'App Depot Lockdown' prevents non-admin users from logging in. This can allow for program updates or in case of a user security issue. Simply check the box to Lockdown AppDepot.
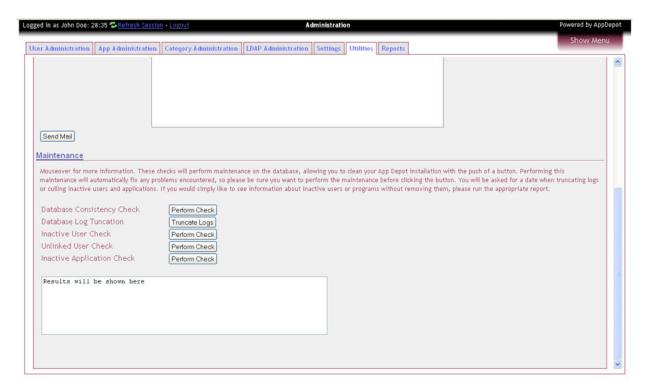
'Communication' uses a form to send email to everyone in the system or to users of a specific application. Use the Application List dropdown to select the users that will be notified.

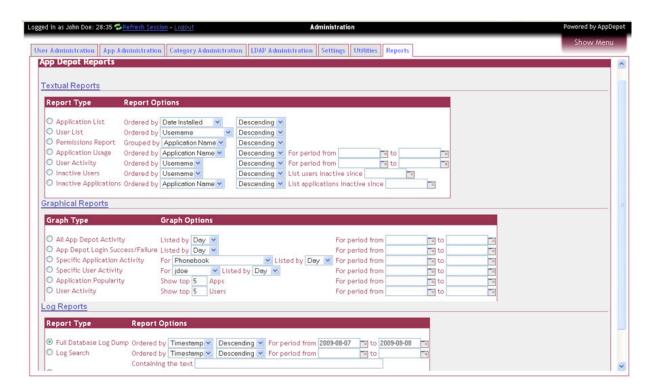Type your email message and click on the 'Send Mail' button.

'Maintenance' checks will perform maintenance on the database, allowing you to clean your App Depot installation with the push of a button. Performing this maintenance will automatically fix any problems encountered, so please be sure you want to perform the maintenance before clicking the button. You will be asked for a date when truncating logs or culling inactive users and applications. If you would simply like to see information about inactive users or programs without removing them, please run the appropriate report.

Reports

Various 'Textual Reports', 'Graphical Reports' or 'Log Reports' options are available as shown on the screen shot below:

Select the appropriate report and click the Fetch Selected Report button:

Bibliography

Michael E. Whitman and Herbert J. Mattord   Principles of Information Security.
Boston, MA: Course Technology, 2009