

mdClassifieds



- Introduction
- Benefits of mdClassifieds
- Benefits of dealing with MDW
- System Topology
- Utility and Application Overview
 - Data Input
 - Data Repository
 - Data Rendering
 - Other Features
- Appendix

Introduction

Readers and advertisers have come to expect Internet classifieds, but fulfillment of that expectation can be costly and time-consuming. Flexible and feature-rich fulfillment can be even more challenging.

mdClassifieds picks up where front-end publishing systems leave off, requiring only a data feed and producing complete, flexible, fully searchable online classifieds listings.

Morris Digital Works' mdClassifieds is the result of four years of continuous technology development and refinement by a media company for a media company.

mdClassifieds answers the challenge of delivering a high volume of dynamically generated search results to a large number of users, while at the same time satisfying the divergent design demands of multiple Web development teams. The system abstracts data from presentation. It delivers data in any text-based format, with equal ease, speed and flexibility. It allows a company's highly-trained database administrators to concentrate on database performance and puts the tools to design and implement data presentation into the hands of the Web design staff.

The purpose of this white paper is to provide an overview of the benefits, requirements and architecture of mdClassifieds.

Benefits of mdClassifieds

The mdClassifieds system supplies developers with all of the necessary tools for capturing, maintaining, and displaying classified ad data. The primary advantages of the mdClassifieds system are:

- **Multiple inputs:** There are several points of data entry into mdClassifieds: Automated loaders parse and insert the large volumes of data generated by publishers' front-end ad entry systems.

mdDisplay facilitates the extraction of text and images from PDF's, so display ads can be presented dynamically - inline with or independent of classified liner ads.

mdClassifieds integrates Web-based Ad Order Entry.

An administrative tool lets newspaper staff add and modify ads live in the mdClassifieds database.

- **Multiple outputs :** Separation of data from presentation allows opportunities for extensive data reuse in a variety of formats and presentations, including effortless aggregation of ads from multiple publications, and repurposing data to exploit new revenue opportunities.
- **Ease of authoring:** Flexible template scripting language provides the ability to modify the look and feel of all data presented from the mdClassifieds suite quickly and easily.
- **Reduced costs:** Most systems require the intervention of highly trained and paid programmers to select, format and display data dynamically. mdClassifieds puts the tools to retrieve and format data into the hands of Web development staff.
- **Security:** mdClassifieds allows users to transfer files easily and securely behind an organization's firewall, and has built-in user account management for controlled access to data modification.
- **Performance:** Most applications or Web pages strive to be data-driven, but the more dynamic a site is, the heavier the requirements are on the database behind it. mdClassifieds incorporates several strategies to address this issue. mdClassifieds applications are written in C, which offers high performance with low overhead and minimal impact on hardware. mdClassifieds also exploits extensive database tuning and data and page caching.

Benefits of dealing with MDW

- **Industry Experience:** mdClassifieds was developed by a media company for a media company. We know publishing because we are publishing. And we've been our own harshest critics and most demanding customers. mdClassifieds is a product we've worked hard on since 1998. And we've gotten it right.
- **Single Vendor Solution:** mdClassifieds contains a wide variety of tools, products, and applications, and supports the gamut of classified needs. The pieces of this package were

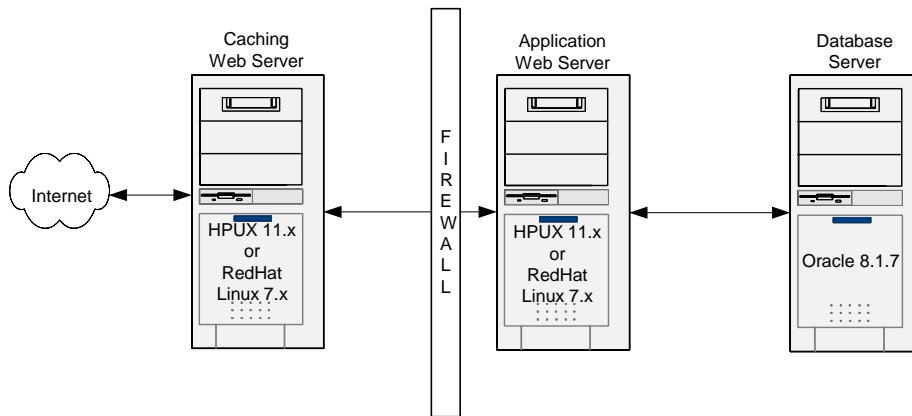
developed to work together.

- **Maturity of Platform:** As business needs have changed, mdClassifieds' has expanded to meet those needs with new applications and functionality. But at its core, mdClassifieds is a product that has been tested continuously in the real world since 1998.

- **Stability:** mdClassifieds was built to answer the needs of Morris' 36 newspapers, 19 shoppers and 20 magazines and specialty publications. MDW has a continuing need for this product and a commitment to its future development and expansion.

System Topology

The next section describes the platform components needed to run mdClassifieds software.



➤ **Recommended server hardware:**

We recommend using Hewlett Packard application and database servers. Specifications will vary. MDW will recommend configurations based on client needs.

- Database server(s)
- Application server(s).
- Although not required, we recommend caching web server(s).

➤ **Server-side system requirements:**

- Oracle 8.1.7 or greater database with InterMedia Text.
- Oracle Net8 client 8.1.7 or greater on all application servers.
- HPUX 11.x or RedHat Linux 7.x
- Apache Web server 2.x

➤ **Recommended server-side components:**

Forge and Oracle contain their own caching mechanisms. In addition, we recommend using Squid Web Proxy Cache and Squirm, a Squid utility. Both pieces of software are free, and open-source.

- Squid caches rendered pages of results, thereby reducing the load on your database. Squid caches based on URL's so results for unique searches will always be returned dynamically, but the database will not need to reprocess the request each time a user browses Merchandise listings, for instance.

➤ **Administrative client-side components:**

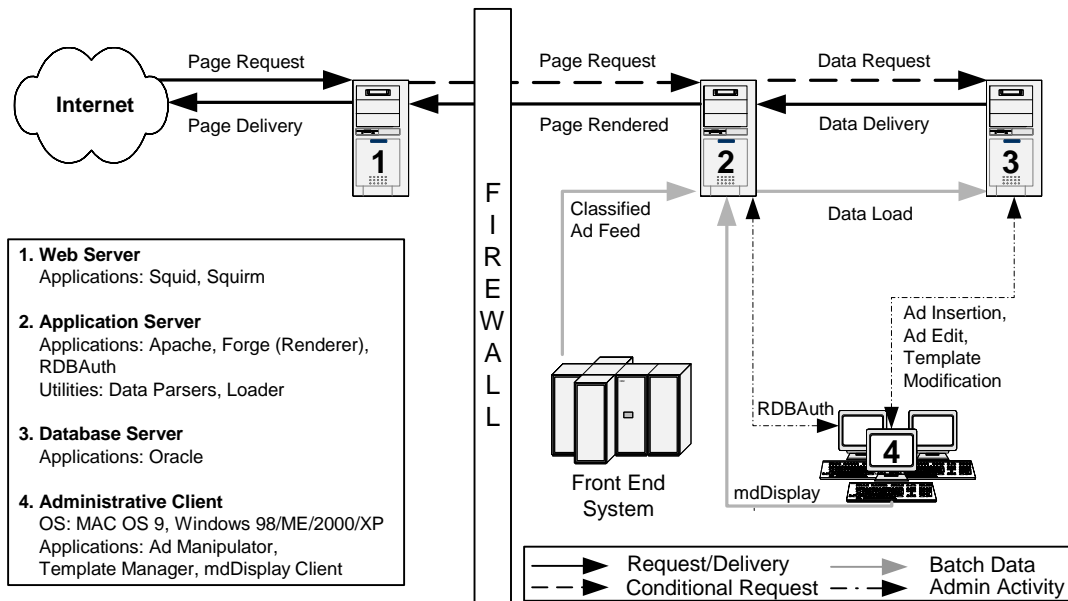
- mdClassifieds' administrative client applications, Template Manager™ and Ad Manipulator, are Java-based. Java runtime (JRE - PC; MRJ - Mac) will be required to run these clients.

Utility/Application Overview

The mdClassifieds system encompasses all of your data input and rendering. The following is a high level view of the mdClassifieds system. Input is built around a loader that accepts classified data from a formatted file. The minimum requirements for this file are the publication dates, classification and text of each ad, but the loader accepts much greater detail. And the richer the data feed, the richer the potential output. Ads can also be input or enhanced through our administrative client and custom versions of Ad Order Entry.

Because mdClassifieds abstracts data from its presentation, output is the centerpiece of our product. Our simple but extremely powerful templating language gives you the ability to deliver content in any text-based format. If you need to deliver content in HTML, XML, comma or tab-delimited text, to a PDA or wireless phone, or to any other format, you will find that you merely need to create a template.

Below is a list of the components comprising the backbone of the mdClassifieds system and their interaction. The overview contains a short description of each of the components and user benefits.



Data Input

➤ Data Parsers, Loader

Ads are sent from the Front End System in a formatted data feed to the Application Server (2), where they're picked up by the Data parsers and Loader. The ads are parsed for feature information, such as the number of bedrooms and bathrooms in a house, or the make, model and mileage of a car (thus expanding the search options available to your users.) The ads and their feature data are then loaded into the MDW classifieds data repository on the Database Server (3).

➤ Ad Manipulator

Ad Manipulator resides on one or more Administrative Client machines (4), and lets classifieds administrators add, edit, or inactivate ads. Users can search for ads or browse categories of ads, then edit any facet of an ad's data, including status, text and run dates. Ad Manipulator includes the ability to manage account creation and access control. Its built-in security keeps users from even seeing ads in properties they don't have access permission for.

➤ mdDisplay

mdDisplay resides on one or more Administrative Client machines (4), and extracts text and images from PDF's. It produces a user-defined number of images. The formatting and sizes of these images are also user-defined. Along with the images, mdDisplay outputs an XML file containing the text of the PDF and a description of the relationship (if any) among the text and images.

The XML is sent to the Data Loader on the Application Server (2), then sent to the Database Server (3). It can then be easily reassembled online to reproduce display ads.

Data Repository

➤ Data Repository

The data repository, on the Database Server (3), contains ad data, indexes for optimized searching capabilities, stored procedures for precompiled data retrieval, automated processes for data archiving, and data modification tracking.

➤ **RDBAuth**

This resides on the Application Server (2) and acts as a security buffer between applications and the database to ensure secure administrative user sessions. Administrative client (4) applications first request authentication from RDBAuth. If they are authenticated, a client/server session proceeds between the Administrative Client (4) and the Database (3).

Data Rendering

➤ **Template Manager**

This resides on one or more Administrative Client machines (4) and allows Webmasters to easily manage a large number of templates written in MTL, MDW's flexible template scripting language (see Appendix for details), by pulling a variety of information from the database.

MTL (pronounced "MeTaL,") can be inserted into any formatting (HTML, XML, etc.) to control presentation. Template output can be displayed dynamically, constantly calling the newest information directly from the database, or generated to static files on a regular basis.

TM provides a quick reference to MTL tags, an audit trail for easy template recovery, and user account management. Template Manager is also equipped with a package containing generic, default templates. These default templates contain examples of all the key functionality offered by MTL and the rendering system.

➤ **Forge**

The Forge rendering engine resides on the Application Server (2) and processes MTL templates into output, conditionally requesting the necessary data from the Data Repository on the Database Server (3) and formatting the returned records.

Forge features data caching to reduce load on the database. If the same data is requested two times in a row, Forge only goes to the database once.

Forge features optimal SQL processing, secure data-entry processing and static and dynamic page generation as well as email generation.

Forge transfers files behind the firewall using RPC protocol. It can deliver files locally or to other hosts in the local network and check whether dependent files exist on those local and remote hosts.

➤ **Internet Requests**

Page requests from the Internet are received by Squid, which is on the Web Server (1.) If Squid has the rendered page for the requested URL in cache, it returns the page immediately to the Web user. If Squid does not have the page cached, it requests the page from Forge, which is on the Application Server (2.)

If Forge has the data required for the page in its cache, it formats the data according to the template's MTL and returns the rendered page to Squid. If Forge does not have the required data, it requests it from the database (3), formats the data according to the template and sends the rendered page to Squid. (Data and presentation are independent of each other, and multiple templates with variously formatted output might call for the same data. So Forge might have the necessary data in cache from an earlier request which output to a different format.)

Squid caches the page returned by Forge and sends it to the Web user.

Other Features

➤ **Ad Order Entry**

Today we have several customized versions of AOE but MDW is in the process of developing a generic template driven e-commerce solution, which promotes substantial additional revenue streams; the Ad Order Entry product (AOE) allows real-time capture of classified ads through an online tool. Due to be released in mdClassifieds v2 (4th Quarter 2002), the AOE tool will allow Web designers to create online submission forms. Classified ad administrators will be able to easily review, edit, and transfer the ads to the live Web site.

➤ **Persistent Search**

This feature of mdClassifieds stores the search criteria of your online classifieds users (e.g. 3-bed, 2-bath home for sale with the keyword "pool.") then notifies users via email when new ads meeting their search criteria become available.

User registration is required for this feature. Email address and the desired notification frequency are the minimum data needed. Other information may be collected and stored at the publisher's discretion.

Appendix

MTL was written to put the tools to retrieve and format data into the hands of proficient HTML writers. It was not targeted to programmers, and has therefore been kept relatively simple. It nonetheless contains the tools to author fairly subtle and complex output.

MTL contains three control structures: if, loop and query. The query structure, which controls the retrieval and output of data, is the heart of MTL.

A simple MTL template:

```
<mtl classifieds.ad property=$property>
  <mtl classifieds.ad.free_text><p>
</mtl classifieds.ad>
```

This example retrieves the first 20 ads from the property or properties specified by `$property`. It consists of an open query tag: `<mtl classifieds.ad property=$property>`, a close query tag: `</mtl classifieds.ad>`, a data tag and an HTML tag.

The open query tag defines the start of the query control structure and the criteria Forge will use to select ads. The close query tag defines the end of the query control structure. Forge executes everything between the open and the close query tags once for each ad that matches the criteria. In this case, Forge will output the text of the ad, followed by a "`<p>`."

Pagination is built in to `mdClassifieds`, so unless the open query tag contains instructions to the contrary, Forge will break the query results into "pages" of 20 ads each.

To be able to get to subsequent pages of ads, or to narrow the selection criteria (from "all ads in `$property`") add more search criteria, also known as *query qualifiers*.

```
<mtl classifieds.ad property=$property ads_per_page=$maxrec
page=$page category_number=$categorynumber
classification=$classification>
  <mtl classifieds.ad.free_text><p>
</mtl classifieds.ad>
```

This version makes use of several variables. Typically, the values are passed to the template via key=value pairs in the URL used to access the template. Here's how arguments look in a URL:

```
.../classifieds-bin/classifieds?property=[your
prop] &maxrec=10&page=2&classification=transportation...
```

Note that no category number has been specified in this URL. Each qualifier in the query is optional. If a value is specified for it, Forge will use it, otherwise it will be ignored.

MTL contains many more query qualifiers and data tags, but a full examination is beyond the scope of this document.