

Administrator's Guide



PUBLISHED BY

Morris DigitalWorks
A Division of Morris Communications LLC

P.O. Box 936
Augusta, GA 30903
Fax: 706-828-4339
Phone: 800-622- 6358
www.morrisdigitalworks.com

Copyright © 2005 By Morris Communications Co., LLC

All Rights Reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the expressed written consent of Morris Communications Corp. Modification of the materials or use of the materials for any other purpose is a violation of Morris Communications Corp.'s proprietary rights.

The product described in this guide is a licensed product of Morris DigitalWorks, LLC. Other brand and product names used herein are for identification purposes only and may be trademarks of their respective owners.

This manual is provided “as is” and without warranties of any kind either expresses or implied. To the fullest extent permissible pursuant to applicable law, Morris Communications disclaims all warranties, express or implied, including but not limited to, implied warranties of merchantability and fitness for a particular purpose. Morris Communications does not warrant that the functions contained in the materials will be uninterrupted or error-free, that defects will be corrected. Morris Communications does not warrant or make any representations regarding the use of or results of the use of the materials in this publications in terms of their correctness, accuracy, reliability or otherwise. You assume the entire cost of all necessary servicing, repair or correction. Applicable law may not allow the exclusion of implied warranties, so the above exclusion may not apply.

Under no circumstances, including, but not limited to, negligence, shall Morris Communications be liable for any special or consequential damages that result from the use of, or the inability to use, the materials in this site, even if an authorized representative has been advised of the possibility of such damages. Applicable law may not allow the limitation or exclusion of liability or incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Revision History

October 2005

Contact Information

For questions or comments regarding this publication contact the Morris DigitalWorks Documentation Department at mdw.doc@morris.com.

Table of Contents

Introduction	2
Providing Feedback.....	2
Reporting Overview	2
Data Collection Process.....	2
Basic Configuration and Setup Tasks.....	3
Database Setup	3
Sequences.....	4
Tables.....	4
Procedures.....	4
Packages.....	4
Data.....	4
Tracking Crumbs	5
Format.....	5
Placement.....	5
Placement Codes.....	8
Crumb Data Storage.....	8
Site Traffic Crumb.....	8
Dealer Search Crumb.....	8
Dealer Details Crumb.....	9
Auto Search Crumb.....	9
Auto Display Crumb.....	9
Dealer Referral Crumb.....	10
Report Logging Server	10
Setup.....	10
Log Rotation.....	12
Special Considerations.....	12

Processor Installation	12
Access Log Processing.....	13
Viewing Reports	14

Introduction

MdTransit's robust web-based reporting system allows site administrators to compile valuable traffic and usage statistics about their Autos site. This document explains how to implement reporting for an autos site. The following topics are covered:

- setting up and managing the system resources necessary for Reporting
- collect data from the autos site pages by implementing tracking crumbs
- explanations of the associated logs, data files
- viewing and understanding final reports

Providing Feedback

To maintain the quality of our publications and software, we welcome your comments on the accuracy, clarity, and value of this publication and all other product documentation. Send comments to the Documentation Department at: mdw.doc@morris.com, or directly to any of the documentation specialists.

Reporting Overview

Reporting data is compiled by tracking the number of image requests made to the reporting server. The image requests (referred to as tracking crumbs) track site activity when they are logged to the Access log and then parsed by the CrumbLog application. After parsing, the crumb data is inserted into the database, then the data is displayed via templates in the DriveTrain application.

Data Collection Process

Below is a description of how data is collected and processed. Each step represents a potential point of failure for the mdTransit reporting system. Items 1-6 represent

unrecoverable failure points.

	Step	Action
Steps 1-6 are unrecoverable failure points	1	User requests mdTransit template.
	2	Template contains tracking crumbs.
	3	The user's browser requests crumb images from the reporting server.
	4	The crumb request is rewritten to a clear gif by a Rewrite rule in the apache configuration.
	5	The request for the crumb is logged in the reporting server's access_log in the CrumbLog format.
	6	The Access log is archived after analog processing.
	7	The archived data file is moved to processing area.
	8	The archived data file is parsed and converted to a loadable format.
	9	The loadable format data file is sorted to help efficiency.
	10	The loadable format data file is loaded into the database.
	11	The Transit Report table is updated to improve efficiency of reports about search behavior.

Note: Steps to ensure that a log file is not processed twice should be taken. Do this by removing the affected data from the database (this must be done manually) and then rerunning data for the affected dates (which typically requires two or more access logs to be merged manually).

Basic Configuration and Setup Tasks

The following list provides a basic overview of the steps that must be performed to deploy and configure the reporting pieces to track data from an autos site.

- Install the necessary files and configure the database.

Note to Enterprise Customers: The information contained in this document assumes that you have been given a .tar.gz file containing binaries, configuration files, etc. If you do not have this file, contact your MDW representative.

- Implement Tracking crumbs- Create the tracking crumb urls and insert them into various web pages within the autos site.
- Setup the Report Logging Server. Configure the CrumbLog to turn logs into counts of key/value pairs
- View Reports. Make any desired modifications to the MTL templates that turn raw counts into end user reports.

Database Setup

This section explains what needs to be setup in your database to run the reporting functionality. It explains what tables need to be created, data, packages, sequences, and packages.

Sequences

MDW_UTIL.mdw_util_seq

Tables

MDW_UTIL.LOG_OBJECT
MDW_UTIL.LOG_OBJECT_ATTRIBUTE
MDW_UTIL.PLACEMENT_CODE
MDW_UTIL.OBJECT_CODE
MDW_UTIL.CUSTOM_TRACKING_ITEM
MDW_UTIL.TRANSIT_REPORT

Procedures

MDW_UTIL.GET_NUMERIC_VALUE
MDW_UTIL.GET_DATE_VALUE

Packages

MDW_UTIL.TRANSIT_REPORT_PKG

Data

MDTRANSIT YMMT category codesets
PLACEMENT_CODES

Tracking Crumbs

This section explains how to implement tracking crumbs to collect autos data for reports, including the correct format and placement of crumbs on site pages.

A crumb is a call for an image (1x1 pixel gif) located within the html code for a web page.

Note: Crumbs may pass additional information through a query string.

Format

All tracking crumbs are in the following format:

```
/image/mdwtc-{$PLACEMENT_CODE}-OBJECT_TYPE-{$OBJECT_ID}?otherdatayouwanttotrack=counts
```

Placement

The following explains where each type of reporting crumb is located within the mdTransit site, who the report data is aimed at (i.e. MBU or dealer) and how to implement the tracking crumb. Refer to the table at the end of this section for the value to insert for the PLACEMENT CODE.

Report Name	Crumb Implementation
Site Traffic	<p>This report is provided to MBUs and provides the hit counts per autos page/page type and count of referring sites. Dealers do not receive this type of data.</p> <p>Place one crumb on each page of the site that passes a placement code as well as the referring domain. It contains an OBJECT_TYPE of MDTRANSIT_SITE and OBJECT_ID of the template property ID (in Template Manager). If the referring hostname is not identical to the current hostname (or nocache.current hostname) a referer_domain key/value should be added to the query string of the image.</p> <p><i>Example:</i></p> <pre>http://rpt.mdtransit.morris.com/images/mdwtc-{\$PLACEMENT_CODE}-MDTRANSIT_SITE-{\$SITE_ID}.gif?referer_domain=domain.com</pre>

Dealer Search

This crumb appears only on the first page of a dealer search results page. It has an OBJECT_TYPE of MDTRANSIT_DEALER_SEARCH and OBJECT_ID of the template property ID (from template manager). The query string of the image contains the following data: make searched for (make=), zip code searched in (zipcode=) and the type of dealer (type=). Types are freeform and what appears in the query string appears on the reports.

Example:

```
http://rpt.mdtransit.morris.com/images/mdwtc-  
${PLACEMENT_CODE}-  
MDTRANSIT_DEALER_SEARCH-  
${SITE_ID}.gif?make=TOYOTA&zipcode=30909&type=(NEW/USED/NEW+AND  
ED
```

Dealer Details

When dealership info is displayed (i.e. dealer searches) a crumb appears with an OBJECT_TYPE of MDTRANSIT_DEALER_DETAILS and OBJECT_ID of the customer ID for that dealer in the classifieds system.

Example:

Enter the following crumb for each dealer:

```
http://rpt.mdtransit.morris.com/images/mdwtc-  
${PLACEMENT_CODE}-MDTRANSIT_DEALER_DETAILS-  
${CUSTOMER_ID}.gif
```

Auto Search

On only the first page of any search results page a crumb appears with an OBJECT_TYPE of MDTRANSIT_II_INVENTORY_SEARCH and OBJECT_ID of the template property ID (in template manager). The query string of the image contains the following data:

- makes searched for (broken out using make.#= syntax) Digits are currently configured to search from 1 to 10.
- canonical model names searched for (i.e. F-Series for ford, not F-150, F-250, F-350) Broken out in the same way as the makes.
- zip codes searched in (zipcode=)
- type of auto (type=).

Types, makes and zip codes are freeform and what appears in the query string appears on the reports. Only canonical model values are detailed below the makes, other values are hidden - canonical values are found in the classifieds codesets category MDTRANSIT, key of MODEL.

Example:

```
http://rpt.mdtransit.morris.com/images/mdwtc-  
${PLACEMENT_CODE}-MDTRANSIT_INVENTORY_SEARCH-  
${SITE_ID}.gif?make.1=TOYOTA&model.1=Camry&zipcode=30909&type  
=(NEW/USED/NEW+AND+USED)
```

The MDTRANSIT codesets category must contain an updated list of makes/models and model aliases for this tracking to work.

Auto Display When auto information is displayed (i.e. inventory search returns; detail pages) a crumb appears with an OBJECT_TYPE of MDTRANSIT_AD and OBJECT_ID of the customer ID for that dealer in the classifieds system. The query string of the ad sends the ad id (ad_id=), make (make=) and model (model=). These are all free form and what appears in the query string is what appears in the reports. Ad ID is not currently stored or displayed.

Example:

```
http://rpt.mdtransit.morris.com/images/mdwtc-
${PLACEMENT_CODE}-MDTRANSIT_AD-
${CUSTOMER_ID}.gif?ad_id=12345&make=TOYOTA&model=CAMRY
```

Dealer Referral

This report is provided to MBUs and provides a count by dealership. Dealers receive reports for the number of times the dealership was linked to.

When linking to a dealer's website, an intermediate page loads containing a crumb with an OBJECT_TYPE of MDTRANSIT_DEALER_URL and OBJECT_ID of the customer ID for that dealer in the classifieds system. To do this, open a JavaScript window that is bugged and has a meta refresh to perform the link.

Example:

```
http://rpt.mdtransit.morris.com/images/mdwtc-
${PLACEMENT_CODE}-MDTRANSIT_DEALER_URL-${CUSTOMER_ID}.gif
```

Inventory Summary

This report data is pulled from live data. It is not necessary to implement a crumb on any of the pages. This report is provided to MBUs and provides the following:

- number of vehicles in inventory (live ads)
- sub-counts by number with photos, price, and exterior color
- Counts by categories of certified dealer, new dealer, used dealer, classifieds.

Dealers can view the following:

- number of vehicles in inventory (live ads)
- sub-counts of number with photos, with price, with exterior color

Dealer Leads

This report provides both MBUs and Dealers with a downloadable count of generated leads. It is not necessary to place a crumb on any of the pages since the data is stored in ad_response. Place the data in KEY=VALUE sequence, and pull back with loop for output to dealer or paper.

Placement Codes

These codes are placed in the various tracking crumbs listed above.

Code	Page	Code	Page
100	Vehicle Results	115	Vehicle Quote Thankyou
101	Vehicle Details	116	Research & Compare
102	Advanced Search	117	Finance Index
103	No Vehicle Results	118	No Dealer Results
104	Full Image	119	E-mail Notification Form
105	Quick Search/Index	120	Quick Quote Form
106	Specification Search	121	E-mail Notification Thankyou
107	Dealer Search	122	Quick Quote Thankyou
108	Specification Results	123	Parts Request
109	Dealer Search Results	124	Service Appointment
110	Comparison	125	Clipboard
111	Comparison Step 3	126	E-mail ad to friend
112	My Selected Vehicles	127	E-mail ad to friend thank you
113	Comparison Step 2	128	Printer Friendly Ad details
114	Calculator		

Crumb Data Storage

This section explains where the mdTransit report data is stored once it is received from the tracking crumb. Each section provides a breakout of the log file pertaining to only that specific report.

Site Traffic Crumb

```
LOG_OBJECTS
  OBJECT_ID = PROPERTY_ID from Template Manager
  OBJECT_TYPE = MDTRANSIT_SITE

LOG_OBJECT_ATTRIBUTE (keys)
  DATE
  DATE HIT_COUNT

  PLACEMENT_CODE
  PLACEMENT_CODE HIT_COUNT

  REFERER_DOMAIN (this must be passed within the image call)
  REFERER_DOMAIN HIT_COUNT
```

Dealer Search Crumb

```
LOG_OBJECTS
  OBJECT_ID = PROPERTY_ID
  OBJECT_TYPE = MDTRANSIT_DEALER_SEARCH
```

```

LOG_OBJECT_ATTRIBUTE
  DATE
    DATE HIT_COUNT
    MAKE
      MAKE HIT_COUNT
    ZIP
      ZIP HIT_COUNT
    TYPE
      TYPE HIT_COUNT

```

Dealer Details Crumb

```

LOG_OBJECTS
  OBJECT_ID = CUSTOMER_ID
  OBJECT_TYPE = MDTRANSIT_CUSTOMER_DETAILS

LOG_OBJECT_ATTRIBUTE (keys)
  DATE
    DATE HIT_COUNT

```

Auto Search Crumb

```

LOG_OBJECTS
  OBJECT_ID = PROPERTY_ID
  OBJECT_TYPE = MDTRANSIT_II_INVENTORY_SEARCH

LOG_OBJECT_ATTRIBUTE
  DATE
    DATE HIT_COUNT
    TYPE
      TYPE HIT_COUNT
    ZIP
      ZIP HIT_COUNT
    MAKE
      MAKE HIT_COUNT
    MODEL
      MODEL HIT_COUNT

```

Auto Display Crumb

```

LOG_OBJECTS
  OBJECT_ID = AD_CUSTOMER_ID
  OBJECT_TYPE = MDTRANSIT_AD

LOG_OBJECT_ATTRIBUTE
  DATE
    DATE HIT_COUNT
    PLACEMENT_CODE
      PLACEMENT_CODE HIT_COUNT

```

```

MAKE
  MAKE HIT_COUNT
MODEL
  MODEL HIT_COUNT

```

Dealer Referral Crumb

```

LOG_OBJECTS
  OBJECT_ID = CUSTOMER_ID
  OBJECT_TYPE = MDTRANSIT_DEALER_URL

LOG_OBJECT_ATTRIBUTE (keys)
  DATE
  DATE HIT_COUNT

```

Report Logging Server

Reporting is accomplished by placing tracking crumbs on site pages. These crumbs transmit information about what pages are being viewed and what is being displayed on each page. The Report Logging Server logs these requests. This section explains the information necessary to configure this server.

Note: A 1 x 1 pixel gif file, a sample server logging configuration file and a sample .htaccess file for the /images directory can be found in the Logging directory of the reporting installation package.

Setup

The report logging server is an Apache server with mod_rewrite enabled, mod_headers enabled, mod_log_config enabled, a rewrite rule, some custom directives to prevent caching of reporting images and a special access log format. Sample configurations for the mod_rewrite module are given below. You may need to adapt these depending on the capabilities of regular expressions on your system.

Directive	Configuration	Discussion	Where to place
-----------	---------------	------------	----------------

LogFormat	LogFormat "%h %{%d-%b-%Y %H %u}t %U %{Referer}i %q" atl	This declares a new log format you can use when specifying where to log information. It logs IP_ADDRESS DATE-MONTH-YEAR HOUR_OF_DAY DAY_OF_WEEK REQUEST REFERER QUERY_STRING	In your virthost directive or main server config
CustomLog	/path/to/log/file/location atl	This will create a log file using the log format necessary for the reporting server	In your virthost directive or main server config
RewriteEngine	on	Enables the rewrite engine	In your virthost directive or main server config
RewriteRule	RewriteRule /images/mdwtc-[0-9]{3}\-([a-z] _ [A-Z])*\-[0-9]*.gif\$ /images/clear.gif	Creates a rewrite rule that will cause any request for an image in the location /images/mdwtc-(3 digits)-(an optional character string with underscores allowed)-(an optional string of digits).gif to be served the image in the location /images/clear.gif	In your virthost directive or main server config
Header	set Cache-Control "max-age=0, no-cache"	This requests that proxy/caching servers not cache this image.	server config, virthost directive or .htaccess file (depending on its location, this directive can affect more than just the clear.gif file)

Header	set Expires "Wed, 31 Dec 2003 12:00:00 GMT"	This requests that the browser not cache this image. The use of this directive is optional and should be based upon your reporting needs.	server config, virthost directive or .htaccess file (depending on its location, this directive can affect more than just the clear.gif file)
--------	---	---	--

Note: A 1 X 1 pixel clear gif file will need to be placed in the following location:
\$DOCUMENT_ROOT/images/clear.gif

Log Rotation

The reporting logs need to be periodically rotated in order for them to be transferred for processing. We recommend that they be rotated every 24 hours. The suggested log format is: \$SERVER_DIRECTORY/logs/rotated/\$Mon/log_name.MMDDYYYY

Special Considerations

If you have . . .	You should . . .
One property (domain names)	Set up one reporting server
Multiple properties (domain names)	Set up your reporting servers so that each domain logs to a separate file

Processor Installation

These instructions assume you are starting with a reporting deployment package tar.gz file

1. Create the root processing directory. It is recommended that this be placed on a segregated volume so that if the filesystem fills up it will not affect other processes. For the duration of this example, we will refer to the reporting root directory as \${PROCESSING_DIRECTORY}

2. Place the reporting deployment package file in the root processing directory and decompress and untar it.

At this point you should have the following directory structure:

- `${PROCESSING_DIRECTORY}`
 - archive
 - incoming
 - bin
 - conf
 - data
 - log
 - parsed
3. Configure the file `${PROCESSING_DIRECTORY}/conf/CrumbLog.cfg` by copying and modifying `${PROCESSING_DIRECTORY}/conf/CrumbLog.cfg.orig`
 4. Configure the `run_nightly.pl`, `sort.sh` and `escaper.pl` files in the bin directory.
 5. Set the environmental variable `CRUMBLOG_HOME` to `${PROCESSING_DIRECTORY}`.
 6. The `run_nightly.pl` program should be cronned to run after `access_log` files are available in the incoming directory. Be sure to allow for files to be fully transferred before starting `run_nightly.pl` process.

Access Log Processing

After image requests are logged to the Access Log the following processing sequence occurs:

Step 1: The Reporting server `access_log` is rotated by customer created process and `access_log` files are placed into the `${PROCESSING_HOME}/incoming` directory. **NOTE:** These files will be deleted during processing.

Step 2: The `run_nightly.pl` program is cronned and will do the following for each file in the incoming directory. Information is logged in the `${PROCESSING_HOME}/log` directory. The log can be used to monitor disk space usage during processing as this information is logged between processing steps.

The Rotated log is transferred for processing to <code>\${ACCESS_LOG_DIRECTORY}</code> (this is set in the <code>CrumbLog.cfg</code> file.)	
Access log is parsed	Run the <code>CrumbLog</code> program with <code>-P</code> option to parse data
Archived access log is removed	All files are removed from <code>\${PROCESSING_HOME}/archive</code> directory.
Parsed file is sorted	Run the <code>sort.sh</code> script to call the <code>sort</code> command on the parsed data file to improve efficiency of load
Parsed file has quotes escaped (mdTransit only)	Run the <code>escaper.pl</code> on sorted file to escape quotes that cause problems during loading.

Line counts are logged	Run <code>wc -l</code> command on the parsed file so that a calculation of the efficiency gained by sort program can be performed.
Parsed file is loaded	Run the CrumbLog program with <code>-L</code> option

Step 3: The Transit Report Table is updated by running the TransitReportUpdate program.

Step 4: Perform filesystem maintenance

Remove files from `${ARCHIVE_DIRECTORY}` (this value is set in the CrumbLog.cfg file). If file archiving is set to false, files must be removed from `${ACCESS_LOG_DIRECTORY}` after each processing run.

Viewing Reports

All reports that are created from the mdTransit data can be viewed through the web-based DriveTrain administrative tool.