Monarch Server Report Mining Edition

Version 14.1

[Developer Guide]

Datawatch Corporation makes no representation or warranties with respect to the contents of this manual or the associated software and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Datawatch Corporation reserves the right to revise this publication and make changes from time to time to its contents without obligation to notify anyone of such revisions or changes.

Monarch Server software is offered and is to be used in accordance with a SOFTWARE LICENSE AND MAINTENANCE AGREEMENT. This agreement stipulates that this software be used only in the computer system designated in that agreement. The agreement further stipulates that the customer shall not copy or alter, or permit others to copy or alter, the software or related materials in whole or in part, in any media for any purpose, except to make an archive (back-up) copy or to make a copy as an essential step in the use of the software with the customer's computer.

Datawatch Corporation hereby grants the buyer the right to reprint this documentation for internal uses only. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, for any other purposes, without the prior written permission of Datawatch Corporation.

DATAWATCH CORPORATION

CORPORATE HEADQUARTERS

4 Crosby Drive
Bedford, MA 01730, USA

Tel.: +1 800.445.3311 / +1 978.441.2200
Investor Relations: +1 978.441.2200 ext. 8323
Fax: 978.441.1114

Sales: sales@datawatch.com

Support: support@datawatch.com

GREENSBORO, NC

101 S Elm #225

Greensboro, NC 27401

Tel: +1 800.445.3311

NEW YORK

415 Madison Avenue, Suite 1421
New York, NY 10017

Tel.: +1 800.445.3311

UNITED KINGDOM

DATAWATCH INTERNATIONAL LIMITED.

Siena Court, Broadway

Maidenhead, Berkshire SL6 1NJ

Tel: +44 845 362 3270

Floor 42 Level 39

One Canada Square

Canary Wharf

London E14 5AB

Tel: +44 845 362 3270

SWEDEN

DATAWATCH AB

Eriksbergsgatan 10

Stockholm, Sweden SE-114 30

Tel: +46 853 480 483

ASIA PACIFIC

MANILA

U2011 20th Flr Jollibee Plaza Condominium

F. Ortigas Jr. Rd., Ortigas Center

Pasig City 1605

PH

Phone +63 2 633 5583

Table of Contents

Introduction

Monarch Server Report Mining Edition (RMS) can be used through an initiation request. The request is sent using the HTTP POST method. It can be generated in a client browser by sending an HTML form with the fields named according to the corresponding parameters of the initiation request.

The initiation request will contain instructions to determine the type of presentation to be provided. The type of presentation can be either a full interactive mining application for the user to select views, or a session-less request to publish a single export of data based on a predefined set of mining rules to a particular data format.

Monarch Server Report Mining Edition Interactive Mining

RMS will recognize the request and then query the report, model, and template data to determine the user's available views.

The initiation request will contain the following information:

- **Initiation Request Type**. A request to provide a full interactive mining application.
- **Report Retrieval information**. This is a complete resolvable path to a source report to be used by RMS.
- **Model Retrieval information**. A complete resolvable path(s) to models to be used by RMS. A folder path is allowed. The folder shall be scanned for .xmod files to obtain a list of all available models.
- **Template Retrieval information** (Excel templates, ES Style style sheets). Provides complete resolvable path(s) to templates to be used by RMS.
- **Model to Template associations**. The templates of types Excel, ES Style shall be associated with a particular model for correct data extraction and processing.
- **The List of Allowed Views**.
- **User ID**. User identifier.
- **Document Type ID**. Server Library Folder identifier.
- **Default View**. The initial view to be presented to the user. The user can navigate to other available views.
- **Dynamic filter**. The initial dynamic filter for Data, Summary and ES Style Views.

Monarch Server Report Mining Edition Export View Request

RMS will recognize the request and then publish the data in the format defined by the initiation request.

The initiation request will contain the following information:

- **Initiation Request Type**. A request to provide a single export of data.
- **Report Retrieval Information**. This is a complete resolvable path to a source report to be used by RMS. Multiple report segments can be specified for processing. They will be concatenated together and then processed.
- **Model Retrieval Information**. A complete resolvable path to a model to be used by RMS.
- **Template Retrieval Information** (Excel templates, ES Style style sheets). Provides complete resolvable path to a template to be used by RMS.
- **Model to Template Associations**. The templates of types Excel, ES Style shall be associated with a particular model for correct data extraction and processing.
- **Data View**. The format of the single export view to be provided to the user.
- **User ID**. User identifier.
- **Dynamic filter**. The initial dynamic filter for Data, Summary and ES Style Views.

Entry Point for Initial Request

The initiation request is sent to the RMS entry point. The URL of entry point is:

```
http://<host name>:<portnumber>/
        <RMS virtual directory>/RequestTypeAnalyze/AnalyzeRequest
```

Where `<host name>` is the name of the computer where the target Monarch Server RMS WebServer component is installed, `<port number>` is the port number on which the RMS WebServer is listening, `< RMS virtual directory>` is the virtual directory under which the application was installed — which is RMSClient by default.

The request is sent using the HTTP POST method. All parameters are passed as the fields of the POST request.

The request's response will either be HTML, PDF, CSV, XLS(X), PRF content or an HTTP error.

Request Parameters Reference

Quick Reference Sheet

Parameter Name	Type M=Mandatory O=Optional		Description
	Interactive	Export	
REQUEST_TYPE	O		Type of request. Valid values are **'I'** for Interactive or **'E'** for a single export view of the mined data. If the type is not specified, the selected type will be Interactive.
REPORT_PATH	M	M	Complete resolvable path to the source report(s). Paths to multiple reports are separated with a semicolon. Can be a local path (accompanied by a host name passed via REPORT_HOST) or an UNC path.
REPORT_HOST	O	O	Name of the host where the report is located. If the report host is not set then the report path must be an UNC path to be equally accessible by any Exporter.
MODEL_PATHS	M	M	Complete resolvable path to the source models (non-template-bound) or a model Definition ID. RMS supports the following formats for this parameter: • path to a single model • list of paths to the models. This list should be in the following format: *<model_1_path>;<model_2_path>;…<model_n_path>* • path to the folder which contains models • model Definition ID This parameter should list only table/summary models and not template-bound models. The template-bound models should be specified in the TEMPLATES_MODELS parameter.

Parameter Name	Type M=Mandatory O=Optional		Description
	Interactive	Export	
TEMPLATE_PATHS	O	O	Complete resolvable path to the source templates or a template Definition ID. RMS supports the following formats for this parameter: • path to a single template • list of paths to the templates. This list should be in the following format: *<template_1_path>;<template _2_path>;… <template_n_path>* • path to the folder which content templates • template Definition ID If this parameter is absent, XFORM data views should not be available.
TEMPLATES_MODELS	O	O	The list of model-to-template associations. Ether paths or object Definition IDs can be used. Parameter format: *<template_1_path>;<model_1_path>;<template _2_path>;<model_2_path>…<template_n_path>;<model_n_path>* If this parameter is absent, XFORM data views should not be available. The template-bound models should be specified in this parameter only and not in MODEL_PATHS.
CHECKSUM	M	M	Contains digital signature for initiation request's parameters.
USER_ID	M	M	User identifier. This identifier used for getting user preferences and for authentication.
USER_PASSWORD	O	O	User password. If a password is not provided, a new dynamic user is created in the RMS DB. This user is assigned to the user group that is specified in the web.config file of the RMS Client. If a password is provided, then in order to login to RMS Client the user with these USER_ID and PASSWORD should already exist and has the RMS User privilege assigned
DOCTYPE_ID	O	N/A	Server Library Folder identifier. If a Server Library Folder identifier is provided, all the models and templates contained in the folder are made available to the user (if the user has the necessary access permissions). This identifier is also used when the user saves a model to the server. If DOCTYPE_ID is not provided, the model cannot be saved to the server. If DOCTYPE_ID is provided, then the model can be stored on the RMS and can be re-used in the next user sessions.

Parameter Name	Type M=Mandatory O=Optional		Description
	Interactive	Export	
DYNAMIC_FILTER	O	O	The initial dynamic filter for Data, Summary and ES Style Views. It consists of a value pair of column name and filter value: *FILTERCOLUMN=[ColumnName];FILTERVALUE=[FilterValue];FILTEROPERATOR=[EQ.NEQ,GT,LT,GTE,LTE,CONTAINS,STARTSWITH,ENDSWITH]* Date format for filter value is YYYYMMDD.
ALLOWED_VIEWS	O	N/A	The list of data views which should be available for user. Parameter format: *<data_view_1 >;<data_view_2>;… < data_view_n>*. Where *<data_view_n>* is one of the following constants: **HTML_REPORT** **DYNAMIC** **TABLE** **SUMMARY** **XLS_TABLE** **XLS_SUMMARY** **PRF** **XFORM** **REMOTE_PORTLETS**
TIMESTAMP	M	M	Timestamp of the request. This is a number of milliseconds elapsed since Jan 01, 1970 (Java-style date/time representation)
JOINPASSWORD	O	O	Join password for logging into external data source.
DEFAULT_VIEW	O	M	Name of the view to show initially. Default is the Welcome page. The view name can be one of the values permitted for the ALLOWED_VIEWS parameter or "WELCOME" value for Welcome page. Parameter is mandatory for Export type request to identify the requested view.

Parameter Name	Type M=Mandatory O=Optional		Description
	Interactive	Export	
EXPORTTO	N/A	M	Defines the format of the exported view. Can be one of the following for HTML_REPORT: **PDF** **HTML** for TABLE: **PDF** **HTML** **CSV** for SUMMARY: **PDF** **HTML** for XLS_TABLE: **XLS** **CSV** for XLS_SUMMARY: **XLS** for PRF: **PRF** for XFORM: **PDF** **HTML**
DELETE_REPORTS	O	O	Report removal flag. Set to "true" or "1" or "on" to remove source report after user session ends.
DELETE_MODELS	O	O	Model removal flag. Set to "true" or "1" or "on" to remove source models after user session ends.
DELETE_TEMPLATES	O	O	Template removal flag. Set to "true" or "1" or "on" to remove source templates after user session ends.
ADMIN	O	N/A	Admin mode flag. Set to "true" or "1" or "on" to allow uploading shared models and deleting models.
REPORT_ENCODING	O	O	Encoding of incoming reports. Can be one of the following: **ANSI** **ASCII** **UTF8** **UTF16LE** **UTF16BE** If the value is not defined or invalid, ANSI will be used.

Parameter Name	Type M=Mandatory O=Optional		Description
	Interactive	Export	
LOCALE	O	O	The locale to be used in the current session. Can be one of the following: **en** (English) **fr-FR** (French-France) **de-DE** (German-Germany) If the value is not defined or invalid, **en** (English) will be used.

Passing Report Paths

The REPORT_PATH should be a local path at the report storage machine and REPORT_HOST is the host name of that machine. The parameter may include path to one report or to multiple reports, in which case the paths must be separated by semicolons. The RMS will access the report(s) directly in the following occasions:

- The Monarch Server Report Mining Edition that is performing the data processing is located at the machine described by the REPORT_HOST parameter.
- REPORT_PATH is an UNC path.

In all other cases the RMS will search for a RmsContentServiceHost component to do report data transfer.

Checksum Generation

The initial request is protected by a checksum — a type of digital signature.

The generation of a checksum includes computing a secure hash for significant attributes of the current request and then signing this hash with a secret key using a XOR operation.

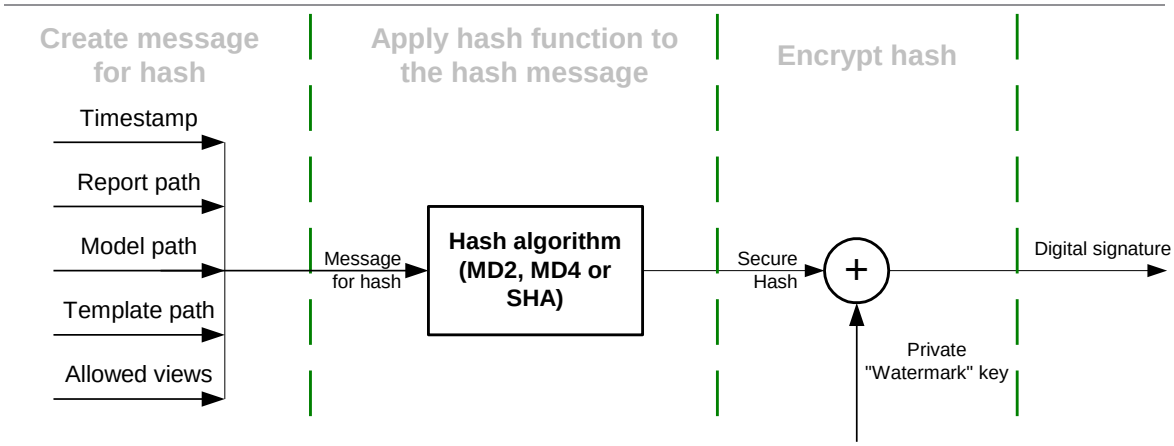

Figure 1. Computing a Checksum

The digital signature should be hex-encoded into symbolic representation before passing into RMS.

CODE EXAMPLE — HASHING A MESSAGE

To create a hash message the request's parameters should be concatenated into one string in the following order: timestamp, report path, model path, template path, allowed views. The code below shows a simple C# function for generating a hash message:

```
// Create message for hash function
        public static String GetHashMessage(
              long timestamp,
              String host,
              String reportPath,
              String modelPath,
              String templatePath,
              String allowedView)
        {
              StringBuilder sb = new StringBuilder();
              // Concantenate all parameters into one String
              sb.Append(timestamp);
              sb.Append(host);
              sb.Append(reportPath);
              sb.Append(modelPath);
              sb.Append(templatePath);
              sb.Append(allowedView);
              // Return result
              return sb.ToString();
        }
```

RMS supports all hash algorithms from the C# library. The algorithms are:

The active algorithm can be set via RMS's configuration file.

Below is a simple C# function for hashing a message:

```
// Hashing
  private static byte[] GetHash(String message, String
hashAlgorithmName)
      {
            byte[] buf = Encoding.UTF8.GetBytes(message);


            // Create hash algorithm instance
            HashAlgorithm algorithm =
HashAlgorithm.Create(hashAlgorithmName);
            // Hash message
            return algorithm.ComputeHash(buf);
      }
```

CODE EXAMPLE — HASH ENCRYPTION

The hash shall be encrypted by an XOR operation with a secret key. The secret key's length is large enough to provide strong encryption. The key can be set differently for every instance of RMS to increase security. It is stored in the RMS configuration file and can be changed by administrator.

Below is shown a simple C# function for hash encryption:

```
        public static String Encrypt(String message, String
watermarkString, String hashAlgorithmName)
        {
              byte[] hash = GetHash(message,
hashAlgorithmName);
              byte[] watermarkKey =
StreamHelper.HexStringToByteArray(watermarkString);
              // Check key Length
              if (hash.Length > watermarkKey.Length)
                    throw new
RmsWebServerException("Incorrect \"watermark\" key.");
              // XOR
              byte[] encryptedHash = new byte[hash.Length];
              for (Int32 i = 0; i < hash.Length; i++)
              {
                    encryptedHash[i] = (byte)(hash[i] ^
watermarkKey[i]);
              }
              // Return result
              return
StreamHelper.ByteArrayToHexString(encryptedHash);
        }    }
```