

MDTest User Manual

Testing Automation and Reporting for IBM i

Midrange Dynamics

Copyright © 2023 Midrange Dynamics

Table of contents

| | |
|--|---|
| 1. MDTTest | 3 |
| 1.1 Welcome to MDTTest | 3 |
| 1.2 MDTTest Installation | 4 |
| 1.2.1 Pre-Requisites | 4 |
| 1.2.2 Where to Install | 4 |
| 1.2.3 Install MDTTest Library | 4 |
| 1.2.4 Cross-Reference MDTEST | 4 |
| 1.2.5 Add MDTEST to Promotion Level Job Descriptions | 4 |
| 1.3 MDTTest Overview | 5 |
| 1.3.1 Other Key Deliverables | 5 |
| 1.3.2 Overall Steps to use MDTTest | 5 |
| 1.4 Defining Test Steps | 6 |
| 1.4.1 Manage Test Definition Categories | 6 |
| 1.4.2 Manage Test Definitions | 6 |
| 1.4.3 Manually Run MDTTest Definition | 8 |
| 1.5 Automated Testing Templates | 9 |
| 1.5.1 Manage Automated Testing Templates | 9 |
| 1.5.2 MDTTest Execution Steps | 0 |
| 1.5.3 Assign MDCMS Attributes to Template | 0 |
| 1.5.4 Object Exceptions | 0 |
| 1.5.5 Batches | 0 |
| 1.6 MDTTest Functions | 0 |
| 1.6.1 Pre-Requisites | 0 |
| 1.6.2 MDTTest Procedures | 0 |
| 1.7 Example Test Programs | 0 |
| 1.7.1 RPG | 0 |
| 1.7.2 COBOL | 0 |
| 1.8 Reporting | 0 |
| 1.8.1 Objects | 0 |
| 1.8.2 MDTTest Jobs | 0 |
| 1.8.3 MDTTest Results | 0 |
| 1.8.4 MDTTest Result Log | 0 |
| 1.8.5 MDTTest Result Log Details | 0 |
| 1.9 Quality Gates | 0 |
| 1.9.1 Override Quality Gate | 0 |

1. MDTest

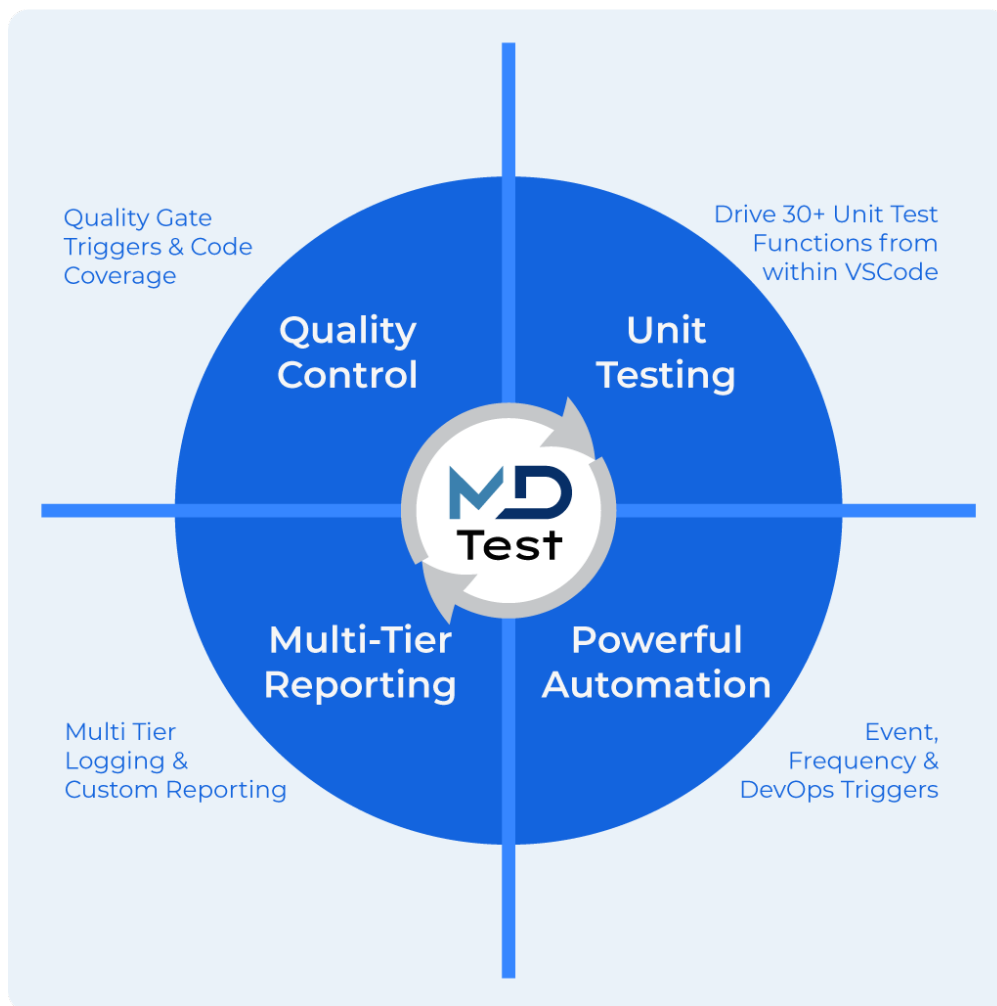
1.1 Welcome to MDTest

This is the reference manual for MDTest - the Midrange Dynamics Automated Testing Tool.

This manual is comprised of these sections:

- [Installation](#)
- [Overview](#)
- [Defining Tests](#)
- [MDTest Functions](#)
- [Example Test Programs](#)
- [Automating Tests](#)
- [Quality Gates](#)
- [Reporting](#)

You can download a PDF version of this guide [here](#).



1.2 MDTest Installation

1.2.1 Pre-Requisites

- MDCMS 8.6.1 or newer
- MDOpen extension for VSCode 8.6+ (available now) or the MDOpen plugin for Eclipse/RDI 8.7+ (under construction)
- License Key for MDTest
- Minimum OS/400 - V7R2M0 TR9
- DevTools - QDEVTOOLS library
- License Program 5770-SS1 39 International Components for Unicode (only required for RegEx testing functions)

1.2.2 Where to Install

MDTest is only necessary to license and install on IBM i partitions where unit testing will occur. This is typically performed on development and functional testing partitions.

1.2.3 Install MDTest Library

A separate download for MDTest isn't necessary, as the installation package for the MDTEST product is included in the MDSEC product library.

A library named MDTEST (+ suffix of the MDCMS instance, when not default) contains all service programs and copybooks necessary for utilizing the MDTest functionality. This library must be installed once before MDTest can be used. Once the library exists, any future MDCMS upgrades run using MDINSSAVF will automatically update the MDTEST library as well.

Command to install the MDTEST library: `MDINSTEST`. The command should be run from a command line from within an MDCMS green-screen session. There are no parameters for this command.

1.2.4 Cross-Reference MDTEST

For ease of use for the developers, it is highly recommended to add the MDTEST library to an application and level in MDXREF and then build the cross-referencing for this library. Steps (for an authorized MDXREF administrator):

1. from green screen, enter command MDXREF and press enter
2. press F9=Build Database
3. select an application/level pair and use option L to list the libraries for the level.
4. position to a blank row and add a sequence number and MDTEST to the library list and press enter.
5. use option X to cross-reference MDTEST

1.2.5 Add MDTEST to Promotion Level Job Descriptions

The job description for each Promotion Level that will utilize MDTest for the compile-time and run-time of test programs will need to have the MDTEST library added to its library list. Steps (for an authorized MDCMS administrator):

1. from green screen, enter command MDCMS and press enter
2. select option 1=MDCMS Setup Menu
3. select option 2=Promotion Levels
4. for each applicable level, use option J=JOB to edit the job description

1.3 MDTTest Overview

MDTest is a Unit testing tool integrated with MDChange.

MDTest provides for the on-demand and automated execution of a sequenced list of commands and SQL statements. The results of the executed items are reported with several layers of granularity and MDTest can block the further deployment of the applicable programs via a quality gate.

Aside from being a runtime engine for testing, MDTest also provides a large set of testing related functions that can be easily imbedded into testing programs written by your development staff (every ILE language is supported). The categories of functions include: - Code Coverage Functions - Performance Measurement Functions - Informational Functions - Execution Functions - Numeric condition Functions - Character condition Functions - RegEx condition Functions - Array condition Functions

[Specifications of all Functions](#)

MDTest functions only need to be embedded in custom test programs - your application objects themselves do not require any special coding.

1.3.1 Other Key Deliverables

- **Powerful Automation**

- Trigger Tests by object, attribute, RFP, period, manually
- Associate with modify, recompile, delete operations
- Invoke any command or SQL statement
- Control Quality Gates with pass/fail results

- **Drill Down Reporting**

- Custom Filters
- Summary Results
- Drill down to individual unit test results
- Custom Reporting Categories
- Tagging, Isolation & Extraction of relevant Job Log data.

- **Reusable Templates**

- Define sequences of Commands or SQL Statements
- Include any pre and post test operations
- Link to MDCMS Attributes
- Generic program inclusion & exclusion filtering
- Use MDCMS Wildcards for runtime variables
- Group granular test definitions
- Pass/Fail Condition Control

1.3.2 Overall Steps to use MDTTest

- [Define Test Steps](#)
- [Write Test Programs](#)
- [Define Automated Testing Templates](#)
- [Reporting](#)
- [Handle Quality Gates](#)

1.4 Defining Test Steps

A testing pipeline is made up of individual steps. In MDTest, a step is represented by a Test Definition.

1.4.1 Manage Test Definition Categories

In MDOpen, navigate to MDTest->MDTest Definition Categories Maintenance of Categories requires authority to MDSEC Code 93

A category is a 10-character identifier of your choosing to help in organizing and filter the list of Test Definitions. Click Add to add a new MDTest Definition Category or click on the category name to edit it.

| Field | Description |
|-------------|---|
| Category | A 10 character ID of your choice to better organize the list of Definitions |
| Description | A suitable description of the category |

| Row Option | Description |
|--------------------|---|
| Copy | Add a new Test Definition Category based on values of an existing element |
| Delete | Delete a Category - only permitted if not used on any definitions |
| Rename | Rename a Category |
| MDTest Definitions | List all definitions for a Category |

1.4.2 Manage Test Definitions

In MDOpen, navigate to MDTest->MDTest Definitions Maintenance of Definitions requires authority to MDSEC Code 93

A Test Definition is a single, executable step that is submitted to batch and processed by the MDTest framework. Each execution of a Definition will be tracked and reported as a Job. A sequence of one or more MDTest Definitions combined into an Automated Testing Template make up a MDTest Pipeline.

Click Add to add a new MDTest Definition or click on the Description to edit it.

| Field | Description |
|--------------------|---|
| Description | A suitable description of the definition |
| Category | A 10 character ID for organizing the Definitions. Use content-assist to select from list of Categories |
| Step Type | Command, Manual or SQL Statement. More information below |
| Timeout in Seconds | Define a maximum wait time (in seconds) for the step to complete before considered failed and to abort the step |
| Step Action | The command string or SQL Statement. Use content-assist to append an MDCMS or Level Wildcard to the string. Step Action is not applicable for manual steps. Use Content-Assist to append a wildcard (MD, Custom Level or Custom Field wildcard) to the Action string. At run-time, MDTest will replace the wildcard with the value based on the target level for level wildcards or the first object request in the RFP that triggered the MDTest automated test. |

| Row Option | Description |
|--------------------------------|---|
| Copy | Add a new Test Definition based on values of an existing definition |
| Delete | Delete a Test Definition - the definition will also be deleted from any Automated Testing Templates |
| MDTest Jobs | List all executions of this definition |
| Manually Run MDTest Definition | Execute the definition manually. Requires MDSEC Code 94. More Details below |
| Templates using Definition | List the Automated Testing Templates that use the MDTest Definition |

Command Step Type

A Command Step Type expects a command in the Step Action parameter. Generally, commands do not need to be qualified with a library name, unless they do not reside in the library list for the job description of the target promotion level. The command will run from within a batch job that exists only for the duration of this command.

Example of using the MDTransform Data copy command to refresh test data:

```
MDCPYDATA TEMP(TESTSETUP) SBM(*NO) JOBS(3)
```

Example of calling a test program with custom wildcard values:

```
CALL PGM(MYTSTPGM) PARM(('##MINCOD##' (*CHAR 5)) ('##MAXCOD##' (*CHAR 5)))
```

Manual Step Type

A manual step is a process that will occur outside of MDTest, such as an interactive process. Once the manual process is complete, a user can click the thumbs-up icon if it passed or thumbs-down icon if it failed. The user and decision will be logged to the step's job as well as the Automated Template batch. The thumb icons will appear in the Test Batches or MDTest Jobs lists for an entry that is waiting for a decision. If a decision isn't granted before the timeout for the step is reached, it will be considered to have failed.

SQL Statement Step Type

An SQL Statement Step Type expects a Db2 SQL statement in the Step Action parameter. This is typically used to insert, update or delete records in a table. The statement will run from within a batch job that exists only for the duration of this statement execution. Generally, SQL statements do not need to be qualified with a library name, unless the target entity doesn't reside in the library list for the job description of the target promotion level.

Example of SQL Statement with a custom wildcard:

```
UPDATE MDACUR SET VALCHF = ++BASRAT++
```

1.4.3 Manually Run MDTest Definition

If a Test Definition should be run manually, to test the definition or to test a target process before/after installing, use option Manually Run MDTest Definition for the selected definition. If the option doesn't appear, then you don't have authority to MDSEC Code 94.

After selecting the option, you will be prompted with a confirmation panel.

| Field | Description |
|------------------------------------|--|
| Appl | The application code of a Promotion Level for determining the job description to use for the library list, job queue, etc. Use content-assist to select from a list |
| Level | The Promotion Level number for determining the job description to use for the library list, job queue, etc. Use content-assist to select from a list |
| Additional Library for top of Libl | The name of a library that should be positioned above all other libraries in the library list. Useful when unit testing a program currently checked out to a developer library |
| Use Additional Library | True = the library will be added to the library list. False = the library will be not be added to the library list |
| Delay Job Start in Seconds | The number of seconds the submitted job should wait before beginning processing. This can be useful for starting a service job in order to debug. The job info is visible from the MDTest Jobs view which is automatically displayed once the Confirm button is pressed. |

1.5 Automated Testing Templates

An Automated Testing Template defines a sequence of Test Definitions that should be executed in order to automatically test an application process. A Template Test is triggered when one or more programs of a given MDCMS attribute have completed installation into a target environment.

1.5.1 Manage Automated Testing Templates

In MDOpen, navigate to MDTest->Automated Testing Templates Maintenance of Testing Templates is permitted if you have authority to either MDSEC Code 7 or 93.

Click Add to add a new Template or click on the template name to edit it.

| Field | Description |
|--------------------|--|
| Template | A 10 character ID of your choice |
| Testing Product | MDTest - the MDTest unit testing product from Midrange Dynamics TestBench - the QA Testing product from Original Software |
| Product Library | If the product is TestBench, the library containing the TestBench product should be entered here |
| Description | A suitable description of the template |
| Quality Gate | When true, any programs that triggered the template will not be allowed to continue beyond the installed level (migration to next level or send to another target), unless the test jobs return a PASS status. The Quality Gate can be overridden by authorized users, when necessary. |
| Test Frequency | The number of candidates found for a template before a batch test run is triggered. The programs will be collected and placed in Test Pending status until the Test Frequency threshold has been reached. The Frequency value is used in conjunction with the Unit of Frequency |
| Unit of Frequency | RFP (recommended, unless test case runs for a long period of time) - A Test Batch will be executed each n (Test Frequency) times an RFP containing one or more candidates is installed Object - A Test Batch will be executed after n object candidates have been installed. If the number of candidates in an RFP exceed the threshold, they will still be in the same batch. Day - A Test Batch will be executed after n days since the first candidate was installed. Manual Only - candidates will be placed in pending status until such time that an authorized user (MDSEC Code 94) manually starts the Test |
| Test Counter | The current number of candidates in pending status |
| Counting from Date | The date of the first currently pending candidate |
| Include Recompiles | By default, only programs checked out for modification are considered candidates. Set to True if Recompiled programs should also trigger a test |
| Include Deletions | Set to True if Deleted programs should trigger a test |
| Include By Default | False - programs installed with an MDCMS attribute assigned to the Template will only be considered a candidate if they are included in the Object Exceptions. True - programs installed with an MDCMS attribute assigned to the Template will always be considered a candidate unless they are omitted in the Object Exceptions. |

| Row Option | Description |
|---------------------------------|---|
| Copy | Add a new Template based on values of an existing element |
| Delete | Delete a Template |
| Rename | Rename a Template |
| Object Exceptions | Since many programs tend to share the same attribute, but may need a different (or no) set of test definitions, individual program handling can be controlled using Object Exceptions. More information below |
| Batches | The list of pending, active or completed executions of the set of Test Definitions used by the Template. More information below. |
| Attributes assigned to Template | The program or service program MDCMS attributes to consider as candidates for testing using this template. |
| MDTest Execution Steps | The sequence of Test Definitions to be executed for the template. More information below. |