

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	11
1.5.3 Assign MDCMS Attributes to Template	11
1.5.4 Object Exceptions	11
1.5.5 Batches	12
1.6 MDTTest Functions	13
1.6.1 Pre-Requisites	13
1.6.2 MDTTest Procedures	13
1.7 Example Test Programs	41
1.7.1 RPG	41
1.7.2 COBOL	42
1.8 Reporting	45
1.8.1 Objects	45
1.8.2 MDTTest Jobs	45
1.8.3 MDTTest Results	45
1.8.4 MDTTest Result Log	46
1.8.5 MDTTest Result Log Details	46
1.9 Quality Gates	47
1.9.1 Override Quality Gate	47

1. MDTest

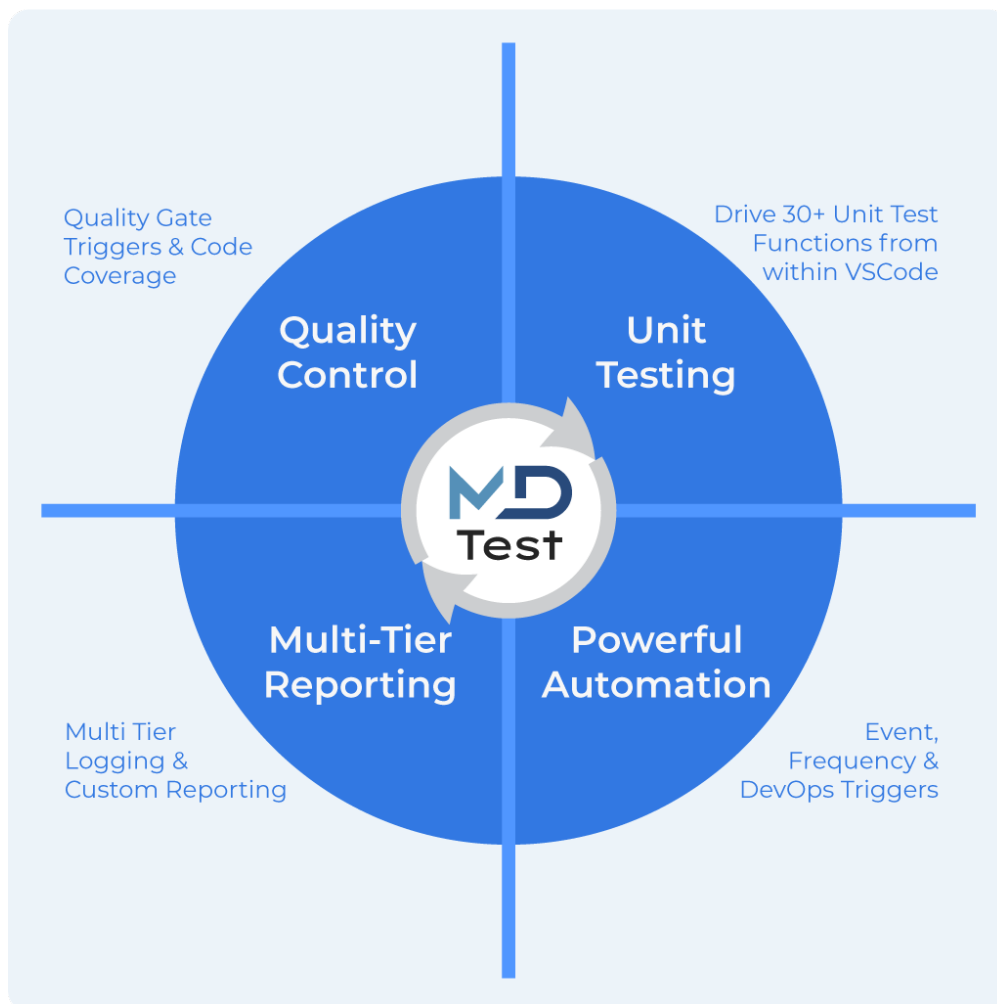
1.1 Welcome to MDTest

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

MDTTest is a Unit testing tool integrated with MDChange.

MDTTest 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 MDTTest can block the further deployment of the applicable programs via a quality gate.

Aside from being a runtime engine for testing, MDTTest 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

MDTTest 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 MDTTest 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, MDTTest 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 MDTTest 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 MDTTest Definition	Execute the definition manually. Requires MDSEC Code 94. More Details below
Templates using Definition	List the Automated Testing Templates that use the MDTTest 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 MDTTest, 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 MDTTest 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 the 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.

1.5.2 MDTest Execution Steps

Viewed/Managed by selecting option MDTest Execution Steps for an Automated Testing Template. Maintenance of Execution Steps requires authority to MDSEC Code 7

An execution step is a sequenced step for the testing pipeline. Each step is executed for a MDTest Template Batch as its own batch job and controlled by the MDTest pipeline manager.

Click Add to add a new Step or click on the sequence number to edit it.

Field	Description
Sequence	A number to order the steps. The lower the number, the earlier the step will be executed.
Definition ID	The ID of a Test Definition. Use content-assist to select from a list
Ignore Errors	false - if the step fails, the entire MDTest batch run will be considered to have failed. true - if the step fails, the batch will be flagged as having warnings but not be set to failed.
Run if Prior Step Failed	false - if any prior step failed, don't execute this step true - execute this step even if a prior step failed (often used for data cleanup steps)

Row Option	Description
Copy	Add a new step based on values of an existing element
Delete	Delete a step
Manually Run MDTest Definition	Run the Definition for the step manually. A confirmation panel will be displayed be execution occurs.

1.5.3 Assign MDCMS Attributes to Template

Viewed/Managed by selecting option Attributes assigned to Template for an Automated Testing Template. Maintenance of Attributes requires authority to MDSEC Code 5 for the target level of the attribute.

Header Option	Description
Only Assigned to This	Only attributes currently assigned to the template will be listed
Not Assigned to This	Only attributes not currently assigned to the template will be listed
Assigned to Any	All attributes will be listed

Row Option	Description
Copy	Add a new step based on values of an existing element
Delete	Delete a step
Manually Run MDTest Definition	Run the Definition for the step manually. A confirmation panel will be displayed before execution occurs.

1.5.4 Object Exceptions

Viewed/Managed by selecting option Object Exceptions for an Automated Testing Template. Maintenance of Object Exceptions is permitted if you have authority to either MDSEC Code 7 or 93.

An Object Exception provides a way to granularly determine how specific programs are tested, as programs for many different purposes will use the same MD attribute.

Click Add to add a new Object Exception or click on the Object Name to edit it.

Field	Description
Object Name	A specific object name or a generic naming pattern for all objects matching the pattern. When MDTest checks the list of Object Exceptions for a program, the specific name will be checked first and then the first matching naming pattern will be used if the specific name isn't found. If a match isn't found in the exception list, then the usage will be determined by the Include by Default parameter on the Automated Testing Template.
Alternative Template	*NONE - do not perform Automated Testing for a matching program *SAME - use the Template that the Object Exception belongs to Alternative Template - use the execution steps defined for a different Template. Use content-assist to select from a list.

Row Option	Description
Copy	Add a new exception based on values of an existing element
Delete	Delete an exception

1.5.5 Batches

Viewed/Managed by selecting option Batches for an Automated Testing Template.

The view lists all batch runs for a specific Testing Template in descending order.

Row Option	Description
Start/Rerun Batch	Start the run for a Batch in Pending status or Rerun a Batch that previously completed or abnormally ended. An initial run or a rerun will both update the quality gate for the batch
Manual Step Passed	If the batch run is waiting for a decision for a manual step, this option is provided to state that the manual step has completed successfully. The batch will then proceed to the next Execution Step.
Manual Step Failed	If the batch run is waiting for a decision for a manual step, this option is provided to state that the manual step has failed. The batch will be flagged as failed (if ignore errors = false) and proceed to the next Execution Step.
Set Batch Status to System Error	If the batch run ended abnormally, this option is used to mark as completed with system errors. Afterwards, the quality gate is set to failed and a subsequent rerun of the batch is possible.
MDTest Jobs	List all of the job information for the batch
Manual Results	List all of the results for the batch
Objects	List all of the installed programs that were included for the batch

1.6 MDTTest Functions

1.6.1 Pre-Requisites

Ensure the prerequisites below are met before adding MDTTest Test Functions described below to your test programs.

COBOL

- In Working Storage Section `COPY MDTCBLWSC OF QLBSLRC.`
- In Procedure Division `COPY MDTCBLPRC OF QLBSLRC.`
- Bind in MDTEST/MDRTEST Service Program (Use any preferred method)
- Ensure MDTEST is in the library list when compiling the program.

RPG

- Include the following copy book: `/Copy QRPGLSRC,MDRTESTP`
- Bind in MDTEST/MDRTEST Service Program (Use any preferred method)
- Ensure MDTEST is in the library list when compiling the program.

1.6.2 MDTTest Procedures

Help Functions

MDTEST_ADDINFO

writes an Information Entry to Result

ILE Procedure Interface

Direction	Field	Type	Options	Comment
In	Message	VarChar(128)	Value	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-ADDINFO USING

Direction	Field	Type	Optional	Initialized	Comment
In	MDTEST-MSG	PIC X(128)	No	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dcl-S h_Message VarChar(128);

//Version 1
h_Message = 'This is my';
MDTest_addInfo(h_Message + ' first message');

//Version 2
MDTest_addInfo('This is my second message':'FINANCE');
```

MDTEST_ADDPASSRESULT

writes a Pass Result

ILE Procedure Interface

Direction	Field	Type	Options	Comment
In	Message	VarChar(128)	Value	
In	Expected Value	VarChar(4096)	Const, NoPass, Omit	
In	Actual Value	VarChar(4096)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-ADDPASSRESULT USING

Direction	Field	Type	Optional	Initialized	Comment
In	MDTEST-MSG	PIC X(128)	No	each Call	
In	MDTEST-EXP-NUM	PIC X(4096)	Yes	each Call	
In	MDTEST-ACT-NUM	PIC X(4096)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Chain 16 CUSTOMER;
If %found(CUSTOMER);
  MDTest_addPassResult('Customer record found.':'Customer!':'exists');
EndIf;
```

MDTEST_ADDLOGENTRY

writes a Log Entry

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Log Entry Id	Int(10)		Input Parameter for MDTest_addLogDetailEntry
In	Message	VarChar(2048)	Value	

COBOL SECTION

MDTEST-ADDLOGENTRY USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-LOG-ID	PIC S9(9) COMP-5			Output Parameter of MDTest_addLogEntry
In	MDTEST-LOG-MSG	PIC X(2048)	No	each Call	

RPG Example

```
Dc1-S h_LogEntryId Like(MDTest_addLogEntry);
h_LogEntryId = MDTest_addLogEntry('Check of Procedure getCustomer finished.');
```

MDTEST_ADDLOGDETAILENTRY

writes a Log Detail Entry

ILE Procedure Interface

Direction	Field	Type	Options	Comment
In	Log Entry Id	Int(10)	Value	Output Parameter of MDTest_addLogEntry
In	Message	VarChar(2048)	Value	

COBOL SECTION

MDTEST-ADDLOGDETAILENTRY USING

Direction	Field	Type	Optional	Initialized	Comment
In	MDTEST-LOG-ID	PIC S9(9) COMP-5	No	first only	Output Parameter of MDTest_addLogEntry
In	MDTEST-LOG-MSG	PIC X(2048)	No	each Call	

RPG Example

```

Dc1-S h_LogEntryId Like(MDTest_addLogEntry);
MDTest_addLogDetailEntry(h_LogEntryId:'Parenter used for calling getCustomer: 16');

```

MDTEST_GETJOBSTATUS

returns the current Job status

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL, MDTest_PASS or MDTEST_UNKNOWN

COBOL SECTION

MDTEST-GETJOBSTATUS USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL, MDTEST-PASS or MDTEST-UNKNOWN

RPG Example

```

Dc1-S h_JobStatus Like(MDTest_getJobStatus);
h_JobStatus = MDTest_getJobStatus();

```

MDTEST_FORCEFAIL

writes a Fail Result

ILE Procedure Interface

Direction	Field	Type	Options	Comment
In	Message	VarChar(128)	Value	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-FORCEFAIL USING

Direction	Field	Type	Optional	Initialized	Comment
In	MDTEST-MSG	PIC X(128)	No	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
MDTest_addInfo('Customer not found.': 'FINANCE');
```

MDTEST_ABORTBATCH

writes a Fail Result

ILE Procedure Interface

Direction	Field	Type	Options	Comment
In	Message	VarChar(128)	Value	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-ABORTBATCH USING

Direction	Field	Type	Optional	Initialized	Comment
In	MDTEST-MSG	PIC X(128)	No	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
MDTest_abortBatch('Customer file not found.': 'FINANCE');
```

MDTEST_WAITMANUALSTEP

Wait for manual actions - Allows for test steps, such as interactive processes, to occur outside of the MDTest framework and then log the test status decision provided by a user from within MDOpen.

ILE Procedure Interface

Direction	Field	Type	Options	Comment
In	Timeout	Int(10)	Const, NoPass, Omit	Maximum wait time in seconds
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-WAIT-MANUAL-STEP USING

Direction	Field	Type	Optional	Initialized	Comment
In	MDTEST-TIMEOUT	PIC S9(9) COMP-5	Yes	each call	
In	MDTEST-REPORT- GROUP	PIC X(10)	Yes	first only	

RPG Example

```
MDTest_waitManualStep(600:'FINANCE');
```

Execute Functions**MDTEST_START_TIMER**

Start the timer for performance measurement and set the start-point for any job log messages to be reported.

After the timer is started, perform one or more program calls, procedure calls, etc. and then use MDTest_End_Timer to stop the timer and collect the results.

When using MDTest_Execute_CMD or MDTest_Execute_SQL, the timer start/stop procedures should be omitted as they are integrated with those 2 execution functions.

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Severity	Packed(2:0)	Value	Severity of Messages to log
In	Duration	Int(10)	Const, NoPass, Omit	Allowed duration in milliseconds between start and end of timer
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-START-TIMER USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-SEV	PIC S9(2) COMP-3	No	first only	Severity of Messages to log
In	MDTEST-MAX- DURATION	PIC S9(9) COMP-5	Yes	each Call	Allowed duration in milliseconds
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT- GROUP	PIC X(10)	Yes	first only	

RPG Example

```
If MDTest_Start_Timer(10:360:'application process':'EXAMPLE') = MDTest_FAIL;
// system error
EndIf;
```

MDTEST_END_TIMER

End the timer that was started using MDTest_Start_Timer and collect any reporting data.

If the duration between the start and end of the timer exceeds the max duration provided to the start procedure, this procedure will return MDTest_FAIL.

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-END-TIMER USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
If MDTest_End_Timer('finished calling my program') = MDTest_FAIL;
// process took too long
EndIf;
```

MDTEST_EXECUTE_CMD

executes an IBM i Command

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Command	VarChar(2048)	Value	
In	Severity	Packed(2:0)	Value	Severity of Messages to log
In	Duration	Int(10)	Const, NoPass, Omit	Allowed duration in milliseconds
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-EXECUTE-CMD USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-CMD	PIC X(2048)	No	each Call	
In	MDTEST-SEV	PIC S9(2) COMP-3	No	first only	Severity of Messages to log
In	MDTEST-MAX-DURATION	PIC S9(9) COMP-5	Yes	each Call	Allowed duration in milliseconds
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
If MDTest_Execute_CMD('ADDLIBLE CUSTDATA':10:360) = MDTest_FAIL;
  //couldnt add CUSTDATA to Library List
EndIf;
```

MDTEST_EXECUTE_SQL

executes an SQL Statement

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Statement	VarChar(2048)	Value	
In	Severity	Packed(2:0)	Value	Severity of Messages to log
In	Duration	Int(10)	Const, NoPass, Omit	Allowed duration in milliseconds
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-EXECUTE-SQL USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-CMD	PIC X(2048)	No	each Call	
In	MDTEST-SEV	PIC S9(2) COMP-3	No	first only	Severity of Messages to log
In	MDTEST-MAX-DURATION	PIC S9(9) COMP-5	Yes	each Call	Allowed duration in milliseconds
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
If MDTest_Execute_SQL('UPDATE CUST set CUSTNAME = '' WHERE CUSTNBR = 16':10:360) = MDTest_FAIL;
//couldnt update field CUSTNAME in file CUST
EndIf;
```

Numeric Functions

MDTEST_NUM_EQUAL

checks if "Expected Value" & "Actual Value" contain the same value

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	Packed(63:23)	Value	
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-EQUAL USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-ACT-NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Num_Equal);
h_Result = MDTest_Num_Equal(CUSTNBR:16:'Customer Nbr should be 16.': 'FINANCE');

```

MDTEST_NUM_NOTEQUAL

checks if "Expected Value" & "Actual Value" contain different values

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	Packed(63:23)	Value	
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-NOTEQUAL USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Num_NotEqual);
h_Result = MDTest_Num_NotEqual(CUSTNBR:16:'Customer Nbr should NOT be 16.': 'FINANCE');

```

MDTEST_NUM_ZERO

checks if "Actual Value" is zero

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-ZERO USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Num_Zero);
h_Result = MDTest_Num_Zero(CUSTNBR);

```

MDTEST_NUM_NOTZERO

checks if "Actual Value" is not zero

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-NOTZERO USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Num_NotZero);
h_Result = MDTest_Num_NotZero(CUSTNBR);

```

MDTEST_NUM_LOWERTHAN

checks if "Actual Value" is lower than "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	Packed(63:23)	Value	
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-LOWERTHAN USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dc1-S h_Result Like(MDTest_Num_LowerThan);
h_Result = MDTest_Num_LowerThan(CUSTNBR:16);

```

MDTEST_NUM_LOWEREQUALTHAN

checks if "Actual Value" is lower than or equal "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	Packed(63:23)	Value	
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-LOWEREQUALTHAN USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dc1-S h_Result Like(MDTest_Num_LowerEqualThan);
h_Result = MDTest_Num_LowerEqualThan(CUSTNBR:16);
```

MDTEST_NUM_GREATERTHAN

checks if "Actual Value" is greater than "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	Packed(63:23)	Value	
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-GREATERTHAN USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Num_GreaterThan);
h_Result = MDTest_Num_GreaterThan(CUSTNBR:16);

```

MDTEST_NUM_GREATEREQUALTHAN

checks if "Actual Value" is greater than or equal "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	Packed(63:23)	Value	
In	Actual Value	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-GREATEREQUALTHAN USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Num_GreaterEqualThan);
h_Result = MDTest_Num_GreaterEqualThan(CUSTNBR:16);

```

MDTEST_NUM_INTRANGE

checks if "Actual Value" is between "Range Start" and "Range End"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	Packed(63:23)	Value	
In	Range Start	Packed(63:23)	Value	
In	Range End	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-INRANGE USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-RANGE- START	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-RANGE- END	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

DC1-S h_Result Like(MDTest_Num_InRange);
h_Result = MDTest_Num_InRange(CUSTNBR:1:229);

```

MDTEST_NUM_OUTSIDERANGE

checks if "Actual Value" is not between "Range Start" and "Range End"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	Packed(63:23)	Value	
In	Range Start	Packed(63:23)	Value	
In	Range End	Packed(63:23)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-NUM-OUTSIDERANGE USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-ACT- NUM	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-RANGE- START	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-RANGE- END	PIC S9(40)V9(23)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST- REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dc1-S h_Result Like(MDTest_Num_OutsideRange);
h_Result = MDTest_Num_OutsideRange(CUSTNBR:1:229);
```

Character Functions

MDTEST_CHAR_EQUAL

checks if "Expected Value" & "Actual Value" contain the same value

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-EQUAL USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dc1-S h_Result Like(MDTest_Char_Equal);
h_Result = MDTest_Char_Equal('MDTest':'MDTEST':*On);
```

MDTEST_CHAR_NOTEQUAL

checks if "Expected Value" & "Actual Value" contain different values

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-NOTEQUAL USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dcl-S h_Result Like(MDTest_Char_NotEqual);
h_Result = MDTest_Char_NotEqual('MDTest':'MDTEST':*On);
```

MDTEST_CHAR_CONTAINS

checks if "Actual Value" contains "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-CONTAINS USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dcl-S h_Result Like(MDTest_Char_Contains);
h_Result = MDTest_Char_Contains('test':'MDTest':*On);
```

MDTEST_CHAR_NOTCONTAINS

checks if "Actual Value" does not contain "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-NOTCONTAINS USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dc1-S h_Result Like(MDTest_Char_NotContains);
h_Result = MDTest_Char_NotContains('test': 'MDTest': *On);
```

MDTEST_CHAR_STARTSWITH

checks if "Actual Value" starts with value of "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-STARTSWITH USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dcl-S h_Result Like(MDTest_Char_Startswith);
h_Result = MDTest_Char_Startswith('MD':'MDTest');
```

MDTEST_CHAR_ENDSWITH

checks if "Actual Value" ends with value of "Expected Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-ENDSWITH USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dc1-S h_Result Like(MDTest_Char_Endswith);
h_Result = MDTest_Char_Endswith('est':'MDTest');
```

MDTEST_CHAR_BLANK

checks if "Actual Value" is empty

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	VarChar(4096)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-BLANK USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dc1-S h_Result Like(MDTest_Char_Blank);
Dc1-S h_Name Char(10) Inz('Sam');
```

```
h_Result = MDTest_Char_Blank(h_Name);
```

MDTEST_CHAR_NOTBLANK

checks if "Actual Value" is not empty

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	VarChar(4096)	Value	
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION
MDTEST-CHAR-NOTBLANK USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
Dcl-S h_Result Like(MDTest_Char_NotBlank);
Dcl-S h_Name Char(10) Inz('Sam');

h_Result = MDTest_Char_NotBlank(h_Name);
```

MDTEST_CHAR_IN

checks if "Actual Value" is found in "Elements"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	VarChar(256)	Value	
In	Number of Elements	Int(10)	Value	
In	Elements	VarChar(256)	Dim(50), Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-IN USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-CHAR-ELEM	PIC X(256)	No	each Call	
In	MDTEST-CARRAY-COUNT	PIC S9(9) COMP-4	No	first only	
In	MDTEST-CHAR-ARRAY	PIC X(256)	No	first only	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```

Dcl-S h_Result Like(MDTest_Char_In);
Dcl-S h_Elements VarChar(256) Dim(50);

h_Elements(1) = 'ABC';
h_Elements(2) = 'DEF';

h_Result = MDTest_Char_In('def':2:h_Elements:*On);

```

MDTEST_CHAR_NOTIN

checks if "Actual Value" is not found in "Elements"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Actual Value	VarChar(256)	Value	
In	Number of Elements	Int(10)	Value	
In	Elements	VarChar(256)	Dim(50), Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-NOTIN USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-CHAR-ELEM	PIC X(256)	No	each Call	
In	MDTEST-CARRAY-COUNT	PIC S9(9) COMP-4	No	first only	
In	MDTEST-CHAR-ARRAY	PIC X(256)	No	first only	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Exempleregex

```

Dcl-S h_Result Like(MDTest_Char_NotIn);
Dcl-S h_Elements VarChar(256) Dim(50);

h_Elements(1) = 'ABC';
h_Elements(2) = 'DEF';

h_Result = MDTest_Char_NotIn('def':2:h_Elements:*On);

```

MDTEST_CHAR_REGEX

checks if Regular Expression in "Expected Values" is found in "Actual Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-REGEX USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
h_Status = MDTest_Char_Regex('^(?:\w+\.)*\w+@(?:\w+\.)*\w+$':'Saman.Neinawaie@MidrangeDynamics.com');
```

MDTEST_CHAR_NOTREGEX

checks if Regular Expression in "Expected Values" is not found in "Actual Value"

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Expected Value	VarChar(4096)	Value	
In	Actual Value	VarChar(4096)	Value	
In	Ignore Case	Ind	Const, NoPass, Omit	Case Insensitive On/Off
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-CHAR-NOTREGEX USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-EXP-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-ACT-CHAR	PIC X(4096)	No	each Call	
In	MDTEST-IGNORE-CASE	PIC 1	Yes	first only	
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
h_Status = MDTest_Char_NoRegex('^(?:\w+\.)?*\w+@(?:\w+\.)*\w+$': 'Saman.Neinawaie@MidrangeDynamics.com');
```

Code Coverage Functions

MDTEST_REG_CODECOV_MOD

register code coverage for a module

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Program Library	Char(10)	Value	Library Name, *LIBL, *CURLIB
In	Program Name	Char(10)	Value	
In	Module	Char(10)	Value	
In	Program Type	Char(10)	Value	*PGM, *SRVPGM
In	min. %	Int(10)	Const, NoPass, Omit	
In	max. %	Int(10)	Const, NoPass, Omit	
In	min. Hit Lines	Int(10)	Const, NoPass, Omit	
In	max. Hit Lines	Int(10)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-REG-CODECOV-MOD USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-PGM-LIB	PIC X(10)	No	first only	
In	MDTEST-PGM-NAME	PIC X(10)	No	first only	
In	MDTEST-PGM-MODULE	PIC X(10)	No	first only	
In	MDTEST-PGM-TYPE	PIC X(10)	No	first only	*PGM, *SRVPGM
In	MDTEST-MIN-PERCENTAGE	PIC S9(9) COMP-5	Yes	each Call	
In	MDTEST-MAX-PERCENTAGE	PIC S9(9) COMP-5	Yes	each Call	
In	MDTEST-MIN-LINES-HIT	PIC S9(9) COMP-5	Yes	each Call	
In	MDTEST-MAX-LINES-HIT	PIC S9(9) COMP-5	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	
##### RPG Example					

```
h_Status = MDTest_reg_CodeCov_Mod('MYLIB':'MYPGM':'*ALL':'*PGM':85:-1:1961:-1:'FINANCE');
```

MDTEST_RPT_CODECOV

generate code coverage report

ILE Procedure Interface

Direction	Field	Type	Options	Comment
Out	Status	Char(10)		MDTest_FAIL or MDTest_PASS
In	Message	VarChar(128)	Const, NoPass, Omit	
In	Report Group	Char(10)	Const, NoPass, Omit	

COBOL SECTION

MDTEST-RPT-CODECOV USING

Direction	Field	Type	Optional	Initialized	Comment
Out	MDTEST-STATUS	PIC X(10)			MDTEST-FAIL or MDTEST-PASS
In	MDTEST-MSG	PIC X(128)	Yes	each Call	
In	MDTEST-REPORT-GROUP	PIC X(10)	Yes	first only	

RPG Example

```
h_Status = MDTTest_rpt_CodeCov();
```


1.7 Example Test Programs

1.7.1 RPG

Below is an example RPG free-format program utilizing the various [MDTest Helper Functions](#)

```

**Free

//*****
//Compile Settings *
//*****
Ctl-Opt Alloc(*Teraspace) Option(*SrcStmt:*NoDebugIO:*NoUnRef)
      Main(mainProc);

//*****
//Copybooks *
//*****
/Copy QRPGLSRC,MDRTESTP

//*****
//Global Variables & Data Structures *
//*****

//*****
//Main Procedure - *ENTRY *
//*****
Dcl-Proc mainProc;
  Dcl-Pi mainProc;
    CmdDur Zoned(6);
    SQLDur Zoned(6);
  End-Pi;

  Dcl-DS d_SQLRow Qualified Dim(100);
    CurCod Char(3);
    Act_FX Packed(8:3);
    Exp_FX Packed(8:3);
  End-DS;

  Dcl-S Cnt Packed(3);
  Dcl-S Date Char(8);

  Dcl-S Status Like(MDTest_reg_CodeCov_Mod);
  Dcl-S Cmd VarChar(2048);
  Dcl-S Elem VarChar(256) Dim(50);
  Dcl-S Row Int(10);
  Dcl-S i Int(10);

  Dcl-C c_Quote Const('');

  //register application program for code coverage
  Status = MDTest_reg_CodeCov_Mod('*LIBL':
    'MDACURUPD':
    '*ALL':
    '*PGM':
    50:
    *Omit:
    15:
    *Omit:
    'FINANCE');

  If Status = MDTest_FAIL;
    Return;
  EndIf;

  //invoke currency update process
  Date = %char(%date():*ISO0);
  Cmd = 'CALL PGM(MDACURUPD) PARM(' + c_Quote +
    Date + c_Quote + ')';
  Status = MDTest_Execute_CMD(
    10:
    CmdDur:
    'Currency update process');

  If Status = MDTest_FAIL;
    Return;
  EndIf;

  //generate code coverage report
  MDTest_rpt_CodeCov('code coverage for currency update pgm');

  Cmd = 'UPDATE MDTEST_COMPARE_RATES A ' +
    'SET A.CHF_FX = (RRN(A) / 10) + 1';
  Status = MDTest_Execute_SQL(Cmd:
    10:
    SQLDur:
    'Generate Test Values');

  If Status = MDTest_FAIL;
    Return;
  EndIf;

```

```

MDTest_addInfo('various tests using the test table');

Exec SQL
  DECLARE C1 CURSOR FOR
  SELECT A.CURCOD, A.VALCHF, E.CHF_FX
  FROM MDACUR A, MDTEST_COMPARE_RATES E
  WHERE A.CURCOD = E.CURCOD
  ORDER BY A.CURCOD;

Exec SQL
  Open C1;

DoW 1 = 1;
Exec SQL
  Fetch C1 for 100 Rows into :d_SQLRow;

If SQLCode <> 0;
  Leave;
EndIf;

For i = 1 to SQLER3;
  Elem(i) = d_SQLRow(i).CurCod;
  Row += 1;

  MDTest_Char_NotRegex('/':d_SQLRow(i).CurCod);

  MDTest_Num_Equal(d_SQLRow(i).Exp_FX:
    d_SQLRow(i).Act_FX:
    d_SQLRow(i).CurCod);

  MDTest_Num_InRange(d_SQLRow(i).Act_FX:
    d_SQLRow(i).Exp_FX - 0,1:
    d_SQLRow(i).Exp_FX + 0,1:
    d_SQLRow(i).CurCod);

EndFor;

EndDo;

Exec SQL
  Close C1;

MDTest_Char_In('EUR':Row:Elem);
MDTest_Char_In('GBP':Row:Elem);

End-Proc;

```

1.7.2 COBOL

2 Copy books are included in the MDTest library that must be included in the source code, in order to take advantage of the [MDTest Functions](#)

1. MDTCBLWSC, to be placed in the WORKING-STORAGE SECTION. Contains all variables used by MDTest. The variables all begin with MDTEST-
2. MDTCBLPRC, to be placed at the end of the PROCEDURE DIVISION. Contains helper section to invoke the MDTest procedures.

```

PROCESS APOST NOMONOPRC
IDENTIFICATION DIVISION.
PROGRAM-ID. CBLEXP1.
AUTHOR. Midrange Dynamics.
DATE-WRITTEN. 20.11.2023.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM1.
OBJECT-COMPUTER. IBM1.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

*****
DATA DIVISION.
FILE SECTION.

*****
WORKING-STORAGE SECTION.

*** MDTest working storage variables
COPY MDTCBLWSC OF QLBSLRC.

EXEC SQL
  INCLUDE SQLCA
END-EXEC.

EXEC SQL
  BEGIN DECLARE SECTION
END-EXEC.

01 SQLTABLE.

```

```

05 SQLROW OCCURS 100.
10 SQLCURCOD PIC X(3).
10 SQLACT-FX PIC S9(5)V9(3) PACKED-DECIMAL.
10 SQLEXP-FX PIC S9(5)V9(3) PACKED-DECIMAL.

EXEC SQL
  END DECLARE SECTION
END-EXEC.

01 WS-CNT          PIC S9(3) COMP-3.
01 WS-DATE        PIC X(8).
01 WS-RETURN-CODE PIC X.
01 WS-ROW         PIC S9(3) COMP-3.

*****
LINKAGE SECTION.
01 LS-CMDDUR.
   05 LS-CMDDUR PIC 9(6).
01 LS-SQLDUR.
   05 LS-SQLDUR PIC 9(6).

PROCEDURE DIVISION USING LS-CMDDUR, LS-SQLDUR.
000-MAIN-CONTROL SECTION.
BEG.

*** register application program for code coverage
MOVE '*LIBL' TO MDTEST-PGM-LIB
MOVE 'MDACURUPD' TO MDTEST-PGM-NAME
MOVE '*ALL' TO MDTEST-PGM-MODULE
MOVE '*PGM' TO MDTEST-PGM-TYPE
MOVE 50 TO MDTEST-MIN-PERCENTAGE
MOVE 15 TO MDTEST-MIN-LINES-HIT
MOVE 'FINANCE' TO MDTEST-REPORT-GROUP
PERFORM MDTEST-REG-CODECOV-MOD

*** skip test suite if unable to register code coverage
IF MDTEST-STATUS = 'FAIL'
  GO TO XIT
END-IF

*** start timer/logging of currency update process
MOVE 10 TO MDTEST-SEV
MOVE LS-CMDDUR TO MDTEST-MAX-DURATION
MOVE 'Currency update process' TO MDTEST-MSG
PERFORM MDTEST-START-TIMER

*** Call application program to perform currency update
ACCEPT WS-DATE FROM DATE YYYYMMDD
CALL 'MYAPPLPGM' USING WS-DATE
                        WS-RETURN-CODE

*** end timer/logging of currency update process
PERFORM MDTEST-END-TIMER

*** generate code coverage report
MOVE 'code coverage for currency update pgm' TO MDTEST-MSG
PERFORM MDTEST-RPT-CODECOV

*** report that process failed if appl program returns E
IF WS-RETURN-CODE = 'E'
  MOVE 'MYAPPLPGM returned value E' TO MDTEST-MSG
  PERFORM MDTEST-FORCEFAIL
  GO TO XIT
END-IF

*** set record values in a hypothetical test table
*** for demonstrating the MDTEST-EXECUTE-SQL
STRING 'UPDATE MDTEST_COMPARE_RATES A '
      'SET A.CHF_FX = (RRN(A) / 10) + 1'
      DELIMITED BY SIZE INTO MDTEST-CMD
MOVE LS-SQLDUR TO MDTEST-MAX-DURATION
MOVE 'Generate Test Values' TO MDTEST-MSG
PERFORM MDTEST-EXECUTE-SQL

*** skip test suite if record update failed
IF MDTEST-STATUS = 'FAIL'
  GO TO XIT
END-IF

*** information message about tests
MOVE 'various tests using the test table'
  TO MDTEST-MSG
PERFORM MDTEST-ADDINFO

*** loop through test table
EXEC SQL
  DECLARE C1 CURSOR FOR
  SELECT A.CURCOD, A.VALCHF, E.CHF_FX
  FROM MDACUR A, MDTEST_COMPARE_RATES E
  WHERE A.CURCOD = E.CURCOD
  ORDER BY A.CURCOD
END-EXEC

EXEC SQL

```

```

OPEN C1
END-EXEC

EXEC SQL
  FETCH C1 FOR 100 ROWS INTO :SQLROW
END-EXEC

IF SQLCODE = 0
  PERFORM MDTEST-CLEAR-CHAR-ARRAY
  MOVE SQLERRD (3) TO WS-CNT
  MOVE 1 TO WS-ROW
  PERFORM 100-COMPARE UNTIL WS-ROW > WS-CNT
END-IF

EXEC SQL
  CLOSE C1
END-EXEC

*** check if EUR in array
MOVE 'EUR' TO MDTEST-CHAR-ELEM
PERFORM MDTEST-CHAR-IN

*** check if GBP in array
MOVE 'GBP' TO MDTEST-CHAR-ELEM
PERFORM MDTEST-CHAR-IN
.

XIT.
GOBACK.

100-COMPARE SECTION.
100.

*** add currency code to array
MOVE SQLCURCOD (WS-ROW) TO MDTEST-CHAR-ELEM
PERFORM MDTEST-ADD-CHAR-ELEM

*** check if currency code contains folder symbol using regex
MOVE '/' TO MDTEST-EXP-CHAR
MOVE SQLCURCOD (WS-ROW) TO MDTEST-ACT-CHAR
PERFORM MDTEST-CHAR-NOTREGEX

*** numeric exact test
MOVE SQLCURCOD (WS-ROW) TO MDTEST-MSG
MOVE SQLEXP-FX (WS-ROW) TO MDTEST-EXP-NUM
MOVE SQLACT-FX (WS-ROW) TO MDTEST-ACT-NUM
PERFORM MDTEST-NUM-EQUAL

*** numeric range test
MOVE SQLCURCOD (WS-ROW) TO MDTEST-MSG
COMPUTE MDTEST-RANGE-START = SQLEXP-FX (WS-ROW) - .1
COMPUTE MDTEST-RANGE-END = SQLEXP-FX (WS-ROW) + .1
MOVE SQLACT-FX (WS-ROW) TO MDTEST-ACT-NUM
PERFORM MDTEST-NUM-INRANGE

ADD 1 TO WS-ROW
.

*** MDTest procedure division sections
COPY MDCBLPRC OF QLBSRC.

```

1.8 Reporting

There are several layers of reporting information that is stored in MDCMS tables when a Test Batch is run.

1.8.1 Objects

View - option Objects for an Automated Testing Batches row

Lists every program object that triggered an automated test run, along with RFP it was in and if a Quality Gate was enabled.

Table - MDCMS.MDDTLUT

Key	Description
MDTEMP	Template Name
MDBCHN	Batch Number
MDRFP	Installed RFP Number
MDOBJN	Object Name
MDOBJL	Object Library
MDOBJT	Object Type

1.8.2 MDTest Jobs

View - option MDTest Jobs for an Automated Testing Batches row

Lists every job (1 per Execution Step) that is running or has run for a batch. When a job is still active, the active job status is displayed and can be clicked on to see the job log for that job.

Table - MDCMS.MDDTJOB

Key	Description
MDTEMP	Template Name
MDBCHN	Batch Number
MDSEQ	Sequence of Template Execution Step (parent = MDCMS.MDDTMDT)
MDTDEF	Definition ID (parent = MDCMS.MDDTDEF)

1.8.3 MDTest Results

View - option MDTest Results for an MDTest Job or Batch row

Lists each result record injected by either your test program or the MDTest program (MDTWRAP). For comparison test results, the actual value and expected value will also be displayed.

Table - MDCMS.MDDTRES

Key	Description
MDJID	The MDTest Job ID (parent = MDCMS.MDDTJOB)
MDTRES	The MDTest Job Result

1.8.4 MDTTest Result Log

View - option MDTTest Result Log for an MDTTest Result row

Lists any upper-level log entries for an injected result record. This will include all options passed to an MDTTest procedure as well as result values up to 128 characters.

Table - MDCMS.MDDTLOG

Key	Description
MDTRES	The MDTTest Job Result ID (parent = MDCMS.MDDTRES)
MDTLID	The MDTTest Job Result Log ID

1.8.5 MDTTest Result Log Details

View - option MDTTest Result Log Details for an MDTTest Result Log row

Lists any lower-level log entries for an injected result record. This will include any included job log entries as well as result values larger than 128 characters.

Table - MDCMS.MDDTLOGD

Key	Description
MDTLID	The MDTTest Job Result Log ID (parent = MDCMS.MDDTLOG)
MDTLSEQ	The MDTTest Job Result Log Detail sequence number

1.9 Quality Gates

After an RFP is installed into a target level, MDCMS checks if any programs that were installed require Automated Testing based on the [Automated Testing Templates](#).

If any programs are included that require testing, then a batch is prepared for each Template that is required.

If the Quality Gate field for one or more of the Automated Testing Templates is set to true, then MDCMS will not allow the submit or send of an RFP containing the programs from the target level until the Test Batch has completed and passed.

Methods to check state of Quality Gate:

- Use option Automated Testing Results for the Installed RFP
- Select function MDTest/All Test Batches from the MDOpen menu

1.9.1 Override Quality Gate

If you have authority to MDSEC Code 59 (Override Automated Testing Quality Gate) for the Level of the installed RFP, you can override the quality gate when the test failed. This is done directly from the RFP pre-submit warning dialog or as an option in the Automated Testing Results view for a Test Batch.