

# Report Banner - Edit rsm.cfg File

Resource Standard Metrics™ for C, C++, C# and Java

Version 7.75 - [mSquaredTechnologies.com](http://mSquaredTechnologies.com)

License Type: Shareware Evaluation License  
Licensed To : Shareware End User - Distribute Freely  
License No. : SW1380 License Date: Dec 05, 1998  
Build Date : Sep 2 2009 Run Date: Apr 22, 2024  
©1996-2009 M Squared Technologies LLC™

License File: C:\Program Files (x86)\MSquared\M2 RSM\rsm.lic  
Config. File: C:\Program Files (x86)\MSquared\M2 RSM\rsm.cfg  
Command Line: -H -OC:\Users\DELL\M2 RSM Wizard\output\functional\_quality  
\_metrics\_report\_for\_databasemanager\_java\_code.htm -c -fa -  
n -o -SFunctional Quality Metrics -FC:\Users\DELL\M2 RSM W  
izard\input\rsm\_file\_list\_for\_databasemanager\_java\_code.ls  
t  
UDQN File : C:\Program Files (x86)\MSquared\M2 RSM\rsm\_udqn.cfg

## Functional Quality Metrics

~~ Function Metrics ~~  
~~ Class/Struct Metrics ~~  
~~ Complexity Analysis ~~  
~~ Quality Analysis ~~

File: [C:\Program Files \(x86\)\MSquared\M2 RSM Wizard\example\\_code\DatabaseManager\DatabaseManager.java](C:\Program Files (x86)\MSquared\M2 RSM Wizard\example_code\DatabaseManager\DatabaseManager.java)

----- Class Begin Line: 8 -----  
Class: DatabaseManager

Notice #52: Line 8: A class has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

Notice #1: Line 9: Line character length = 82. This width exceeds the standard terminal width of 80 characters.

----- Function Begin Line: 13 -----  
Function: DatabaseManager.main  
Parameters: (String[] args)

Notice #51: Line 13: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

Notice #1: Line 66: Line character length = 87. This width exceeds the standard terminal width of 80 characters.

Notice #17: Function comments, 0.0% are less than 10.0%.

Notice #46: Function blank line percent, 0.0% is less than 10.0%.

Notice #28: The function cyclomatic complexity of 13 exceeds the specified limit of 10.

Function: DatabaseManager.main  
Complexity Param 1 Return 1 Cyclo Vg 13 Total 15  
LOC 58 eLOC 56 lLOC 41 Comment 0 Lines 58  
----- Function End Line: 70 -----

----- Function Begin Line: 72 -----  
Function: DatabaseManager.getConnection  
Parameters: ()

**Notice** #51: Line 72: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

**Notice** #49: The function contains no input parameters or void. Suggest using explicit parameters for interface clarity.

Function: **DatabaseManager.getConnection**

Complexity	Param 0	Return 1	Cyclo Vg 1	Total	2
LOC 3	eLOC 2	lLOC 1	Comment 0	Lines	3

----- Function End Line: 74 -----

----- Function Begin Line: 76 -----

Function: **DatabaseManager.addUser**

Parameters: (**Scanner scanner**)

**Notice** #51: Line 76: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

**Notice** #1: Line 83: Line character length = 103. This width exceeds the standard terminal width of 80 characters.

**Notice** #38: Line 83: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

**Notice** #17: Function comments, 6.7% are less than 10.0%.

**Notice** #46: Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.addUser**

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 14	eLOC 14	lLOC 12	Comment 1	Lines	14

----- Function End Line: 89 -----

----- Function Begin Line: 94 -----

Function: **DatabaseManager.updateUser**

Parameters: (**Scanner scanner**)

**Notice** #51: Line 94: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

**Notice** #1: Line 101: Line character length = 103. This width exceeds the standard terminal width of 80 characters.

**Notice** #38: Line 101: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

**Notice** #17: Function comments, 6.7% are less than 10.0%.

**Notice** #46: Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.updateUser**

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 14	eLOC 14	lLOC 12	Comment 1	Lines	14

----- Function End Line: 107 -----

----- Function Begin Line: 112 -----

Function: **DatabaseManager.deleteUser**

Parameters: (**Scanner scanner**)

**Notice** #51: Line 112: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

- Notice** #1: Line 116: Line character length = 103. This width exceeds the standard terminal width of 80 characters.
- Notice** #38: Line 116: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.
- Notice** #17: Function comments, 0.0% are less than 10.0%.
- Notice** #46: Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.deleteUser**

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 10	eLOC 10	lLOC 8	Comment 0	Lines	10

----- Function End Line: 121 -----

----- Function Begin Line: 126 -----

Function: **DatabaseManager.viewUsers**

Parameters: ()

- Notice** #51: Line 126: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
- Notice** #1: Line 128: Line character length = 140. This width exceeds the standard terminal width of 80 characters.
- Notice** #38: Line 128: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.
- Notice** #1: Line 134: Line character length = 84. This width exceeds the standard terminal width of 80 characters.
- Notice** #17: Function comments, 0.0% are less than 10.0%.
- Notice** #46: Function blank line percent, 0.0% is less than 10.0%.
- Notice** #49: The function contains no input parameters or void. Suggest using explicit parameters for interface clarity.

Function: **DatabaseManager.viewUsers**

Complexity	Param 0	Return 1	Cyclo Vg 2	Total	3
LOC 11	eLOC 10	lLOC 8	Comment 0	Lines	11

----- Function End Line: 136 -----

----- Function Begin Line: 141 -----

Function: **DatabaseManager.addRole**

Parameters: (Scanner scanner)

- Notice** #51: Line 141: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
- Notice** #1: Line 146: Line character length = 103. This width exceeds the standard terminal width of 80 characters.
- Notice** #38: Line 146: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.
- Notice** #17: Function comments, 8.3% are less than 10.0%.
- Notice** #46: Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.addRole**

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 11	eLOC 11	lLOC 9	Comment 1	Lines	11
----- Function End Line: 151 -----					

----- Function Begin Line: 156 -----  
Function: DatabaseManager.assignRole  
Parameters: (Scanner scanner)

- Notice** #51: Line 156: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
- Notice** #1: Line 162: Line character length = 103. This width exceeds the standard terminal width of 80 characters.
- Notice** #38: Line 162: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.
- Notice** #17: Function comments, 0.0% are less than 10.0%.
- Notice** #46: Function blank line percent, 0.0% is less than 10.0%.

Function: DatabaseManager.assignRole

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 13	eLOC 13	lLOC 11	Comment 0	Lines	13
----- Function End Line: 168 -----					

----- Function Begin Line: 173 -----  
Function: DatabaseManager.removeRole  
Parameters: (Scanner scanner)

- Notice** #51: Line 173: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
- Notice** #1: Line 179: Line character length = 103. This width exceeds the standard terminal width of 80 characters.
- Notice** #38: Line 179: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.
- Notice** #17: Function comments, 0.0% are less than 10.0%.
- Notice** #46: Function blank line percent, 0.0% is less than 10.0%.

Function: DatabaseManager.removeRole

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 13	eLOC 13	lLOC 11	Comment 0	Lines	13
----- Function End Line: 185 -----					

----- Function Begin Line: 190 -----  
Function: DatabaseManager.listRoles  
Parameters: ()

- Notice** #51: Line 190: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.
- Notice** #1: Line 192: Line character length = 140. This width exceeds the standard terminal width of 80 characters.
- Notice** #38: Line 192: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

**Notice #1:** Line 197: Line character length = 84. This width exceeds the standard terminal width of 80 characters.

**Notice #17:** Function comments, 0.0% are less than 10.0%.

**Notice #46:** Function blank line percent, 0.0% is less than 10.0%.

**Notice #49:** The function contains no input parameters or void. Suggest using explicit parameters for interface clarity.

Function: **DatabaseManager.listRoles**

Complexity	Param 0	Return 1	Cyclo Vg 2	Total	3
LOC 10	eLOC 9	lLOC 7	Comment 0	Lines	10

----- Function End Line: 199 -----

----- Function Begin Line: 204 -----

Function: **DatabaseManager.searchUserByName**

Parameters: (**Scanner scanner**)

**Notice #51:** Line 204: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

**Notice #1:** Line 209: Line character length = 103. This width exceeds the standard terminal width of 80 characters.

**Notice #38:** Line 209: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

**Notice #38:** Line 211: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

**Notice #1:** Line 216: Line character length = 88. This width exceeds the standard terminal width of 80 characters.

**Notice #17:** Function comments, 6.3% are less than 10.0%.

**Notice #46:** Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.searchUserByName**

Complexity	Param 1	Return 1	Cyclo Vg 2	Total	4
LOC 15	eLOC 13	lLOC 10	Comment 1	Lines	15

----- Function End Line: 218 -----

----- Function Begin Line: 224 -----

Function: **DatabaseManager.searchUserByEmail**

Parameters: (**Scanner scanner**)

**Notice #51:** Line 224: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

**Notice #1:** Line 229: Line character length = 103. This width exceeds the standard terminal width of 80 characters.

**Notice #38:** Line 229: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

**Notice #38:** Line 231: Exception handling 'try' - 'catch' has been identified. Exception handling can be a form indirect logic control similar to a goto. The use of exception handling should be driven by the design and not casually imposed upon the code.

upon the code.

**Notice #1:** Line 236: Line character length = 88. This width exceeds the standard terminal width of 80 characters.

**Notice #17:** Function comments, 6.3% are less than 10.0%.

**Notice #46:** Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.searchUserByEmail**

Complexity	Param 1	Return 1	Cyclo Vg 2	Total	4
LOC 15	eLOC 13	lLOC 10	Comment 1	Lines	15

----- Function End Line: 238 -----

**Notice #1:** Line 244: Line character length = 107. This width exceeds the standard terminal width of 80 characters.

----- Function Begin Line: 244 -----

Function: **DatabaseManager.executeUpdate**

Parameters: (**PreparedStatement pstmt**, **String successMessage**)

**Notice #51:** Line 244: A function has been identified which does not have a preceding comment. Comments that detail the purpose, algorithms, and parameter/return definitions are suggested.

**Notice #17:** Function comments, 0.0% are less than 10.0%.

**Notice #46:** Function blank line percent, 0.0% is less than 10.0%.

Function: **DatabaseManager.executeUpdate**

Complexity	Param 2	Return 1	Cyclo Vg 2	Total	5
LOC 9	eLOC 7	lLOC 4	Comment 0	Lines	9

----- Function End Line: 252 -----

**Notice #31:** Class comments, 2.1% are less than 10.0%.

**Notice #46:** Class blank line percent, 5.3% is less than 10.0%.

Class: **DatabaseManager**

Attributes	Publ 0	Prot 0	Private 3	Total	3
Methods	Publ 1	Prot 0	Private 22	Total	23
LOC 233	eLOC 201	lLOC 157	Comment 5	Lines	245

----- Class End Line: 253 -----

**Notice #47:** File blank line percentage, 5.9% is less than 10.0%

**Notice #20:** File comment line percentage, 2.0% is less than 10.0%

~~ Total File Summary ~~

LOC 239	eLOC 207	lLOC 163	Comment 5	Lines	253
---------	----------	----------	-----------	-------	-----

-----

~~ File Functional Summary ~~

File Function Count....:	13			
Total Function LOC.....:	196	Total Function Pts LOC :		4.5
Total Function eLOC.....:	185	Total Function Pts eLOC:		3.9
Total Function lLOC.....:	144	Total Function Pts lLOC:		3.1
Total Function Params ..:	11	Total Function Return ..:		13
Total Cyclo Complexity :	30	Total Function Complex.:		54
-----	-----	-----	-----	-----
Max Function LOC .....	58	Average Function LOC ..:		15.08
Max Function eLOC .....	56	Average Function eLOC ..:		14.23
Max Function lLOC .....	41	Average Function lLOC ..:		11.08
-----	-----	-----	-----	-----
Max Function Parameters:	2	Avg Function Parameters:		0.85
Max Function Returns ..:	1	Avg Function Returns ..:		1.00
Max Interface Complex. :	3	Avg Interface Complex. :		1.85
Max Cyclomatic Complex.:	13	Avg Cyclomatic Complex.:		2.31
Max Total Complexity ..:	15	Avg Total Complexity ..:		4.15

End of File: [C:\Program Files \(x86\)\MSquared\M2 RSM Wizard\example code\DatabaseManager\DatabaseManager.java](#)

~~ Total Metrics For 1 Files ~~

~~ Project Directory Summary ~~

Files 1 - C:\Program Files (x86)\MSquared\M2 RSM Wizard\example code\DatabaseManager  
LOC 239      eLOC 207      lLOC 163      Comment 5      Lines      253

~~ Total Project Summary ~~

LOC 239      eLOC 207      lLOC 163      Comment 5      Lines      253  
Average per File, metric/1 files  
LOC 239      eLOC 207      lLOC 163      Comment 5      Lines      253

~~ Project Functional Metrics ~~

[Function: DatabaseManager.main](#)

Parameters: (String[] args)

Complexity	Param 1	Return 1	Cyclo Vg 13	Total	15
LOC 58	eLOC 56	lLOC 41	Comment 0	Lines	58

[Function: DatabaseManager.getConnection](#)

Parameters: ()

Complexity	Param 0	Return 1	Cyclo Vg 1	Total	2
LOC 3	eLOC 2	lLOC 1	Comment 0	Lines	3

[Function: DatabaseManager.addUser](#)

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 14	eLOC 14	lLOC 12	Comment 1	Lines	14

[Function: DatabaseManager.updateUser](#)

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 14	eLOC 14	lLOC 12	Comment 1	Lines	14

[Function: DatabaseManager.deleteUser](#)

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 10	eLOC 10	lLOC 8	Comment 0	Lines	10

[Function: DatabaseManager.viewUsers](#)

Parameters: ()

Complexity	Param 0	Return 1	Cyclo Vg 2	Total	3
LOC 11	eLOC 10	lLOC 8	Comment 0	Lines	11

[Function: DatabaseManager.addRole](#)

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 11	eLOC 11	lLOC 9	Comment 1	Lines	11

[Function: DatabaseManager.assignRole](#)

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 13	eLOC 13	lLOC 11	Comment 0	Lines	13

[Function: DatabaseManager.removeRole](#)

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 1	Total	3
LOC 13	eLOC 13	lLOC 11	Comment 0	Lines	13



#### Function: DatabaseManager.listRoles

Parameters: ()

Complexity	Param 0	Return 1	Cyclo Vg 2	Total	3
LOC 10	eLOC 9	lLOC 7	Comment 0	Lines	10

#### Function: DatabaseManager.searchUserByName

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 2	Total	4
LOC 15	eLOC 13	lLOC 10	Comment 1	Lines	15

#### Function: DatabaseManager.searchUserByEmail

Parameters: (Scanner scanner)

Complexity	Param 1	Return 1	Cyclo Vg 2	Total	4
LOC 15	eLOC 13	lLOC 10	Comment 1	Lines	15

#### Function: DatabaseManager.executeUpdate

Parameters: (PreparedStatement pstmt, String successMessage)

Complexity	Param 2	Return 1	Cyclo Vg 2	Total	5
LOC 9	eLOC 7	lLOC 4	Comment 0	Lines	9

Total: Functions

LOC 196	eLOC 185	lLOC 144	InCmp 24	CycloCmp	30
---------	----------	----------	----------	----------	----

Function Points	FP(LOC) 3.7	FP(eLOC) 3.5	FP(lLOC)	2.7
-----------------	-------------	--------------	----------	-----

### ~~ Project Functional Analysis ~~

Total Functions .....	13	Total Physical Lines ...	196
Total LOC .....	196	Total Function Pts LOC :	3.7
Total eLOC .....	185	Total Function Pts eLOC:	3.5
Total lLOC.....	144	Total Function Pts lLOC:	2.7
Total Cyclomatic Comp. :	30	Total Interface Comp. ..	24
Total Parameters .....	11	Total Return Points ...:	13
Total Comment Lines ....	5	Total Blank Lines .....	0

Avg Physical Lines ....:	15.08		
Avg LOC .....	15.08	Avg eLOC .....	14.23
Avg lLOC .....	11.08	Avg Cyclomatic Comp. ...:	2.31
Avg Interface Comp. ....:	1.85	Avg Parameters .....	0.85
Avg Return Points .....	1.00	Avg Comment Lines .....	0.38

Max LOC .....	58		
Max eLOC .....	56	Max lLOC .....	41
Max Cyclomatic Comp. ...:	13	Max Interface Comp. ....:	3
Max Parameters .....	2	Max Return Points .....	1
Max Comment Lines .....	1	Max Total Lines .....	58

Min LOC .....	3		
Min eLOC .....	2	Min lLOC .....	1
Min Cyclomatic Comp. ...:	1	Min Interface Comp. ....:	1
Min Parameters .....	0	Min Return Points .....	1
Min Comment Lines .....	0	Min Total Lines .....	3

### ~~ Project Class/Struct Metrics ~~

#### Class: DatabaseManager

Attributes	Publ 0	Prot 0	Private 3	Total	3
Methods	Publ 1	Prot 0	Private 22	Total	23
Complexity	Param 11	Return 13	Cyclo Vg 30	Total	54
LOC 233	eLOC 201	lLOC 157	Comment 5	Lines	245

Total: All Classes/Structs

Attributes	Publ 0	Prot 0	Private 3	Total	3
Methods	Publ 1	Prot 0	Private 22	Total	23
Complexity	Param 11	Return 13	Cyclo Vg 30	Total	54
LOC 233	eLOC 201	lLOC 157	Comment 5	Lines	245



~~ Project Class/Struct Analysis ~~

Total Classes/Structs ..	1	Total Methods .....	23
Total Public Methods ..	1	Total Public Attributes:	0
Total Protected Methods:	0	Total Protected Attrib.:	0
Total Private Methods ..	22	Total Private Attrib. ..	3
Total Physical Lines ..	245	Total LOC .....	233
Total eLOC .....	201	Total lLOC .....	157
Total Cyclomatic Comp. :	30	Total Interface Comp. ..	24
Total Parameters .....	11	Total Return Points ....	13
Total Comment Lines ....	5	Total Blank Lines .....	13
<hr/>			
Avg Physical Lines .....	245.00	Avg Methods .....	23.00
Avg Public Methods .....	1.00	Avg Public Attributes ..	0.00
Avg Protected Methods ..	0.00	Avg Protected Attrib. ..	0.00
Avg Private Methods ....	22.00	Avg Private Attributes :	3.00
Avg LOC .....	233.00	Avg eLOC .....	201.00
Avg lLOC .....	157.00	Avg Cyclomatic Comp. ...	30.00
Avg Interface Comp. ....	24.00	Avg Parameters .....	11.00
Avg Return Points .....	13.00	Avg Comment Lines .....	5.00
<hr/>			
Max Physical Lines .....	245	Max Methods .....	23
Max Public Methods .....	1	Max Public Attributes ..	0
Max Protected Methods ..	0	Max Protected Attrib. ..	0
Max Private Methods ....	22	Max Private Attributes :	3
Max LOC .....	233	Max eLOC .....	201
Max lLOC .....	157	Max Cyclomatic Comp. ...	30
Max Interface Comp. ....	24	Max Parameters .....	11
Max Return Points .....	13	Max Comment Lines .....	5
<hr/>			
Min Physical Lines .....	245	Min Methods .....	23
Min Public Methods .....	1	Min Public Attributes ..	0
Min Protected Methods ..	0	Min Protected Attrib. ..	0
Min Private Methods ....	22	Min Private Attributes :	3
Min LOC .....	233	Min eLOC .....	201
Min lLOC .....	157	Min Cyclomatic Comp. ...	30
Min Interface Comp. ....	24	Min Parameters .....	11
Min Return Points .....	13	Min Comment Lines .....	5

~~ Project Quality Profile ~~

Type Count Percent Quality Notice

1	17	22.67	Physical line length > 80 characters
17	12	16.00	Function comment content less than 10.0%
20	1	1.33	File comment content < 10.0%
28	1	1.33	Cyclomatic complexity > 10
31	1	1.33	Class/Struct comments are < 10.0%
38	12	16.00	Exception Handling "try"- "catch" has been identified
46	13	17.33	Function/Class Blank Line content less < 10.0%
47	1	1.33	File Blank Line content < 10.0%
49	3	4.00	Function appears to have null or blank parameters
51	13	17.33	No comment preceding a function block
52	1	1.33	No comment preceding a class block

75 100.00 Total Quality Notices

~~ Quality Notice Density ~~

Basis: 1000 (K)

Quality Notices/K LOC	=	313.8 ( 31.38%)
Quality Notices/K eLOC	=	362.3 ( 36.23%)
Quality Notices/K lLOC	=	460.1 ( 46.01%)

~~ File Summary ~~

C Source Files *.c ....:	0	C/C++ Include Files *.h:	0
C++ Source Files *.c* ..:	0	C++ Include Files *.h* :	0
C# Source Files *.cs ..:	0	Java Source File *.jav*:	1
Other Source Files ....:	0		
Total File Count .....	1		

Shareware evaluation licenses process only **20** files.  
Paid licenses enable processing for an unlimited number of files.

---

***Report Banner - Edit rsm.cfg File***