CTX-Quality-Control User Guide

Contents

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
10/10/2019	1.0	First Release

Module Versions

The following revisions have been made to this document

Date	Revision	Notes
10/10/2019	1.0	First Release

Preface

About this Manual

This document is a user guide for the CTX-Quality-Control module.

Audience

The audience for this document is those wanting to understand how to use CTX-Quality-Control module.

Related Material

Document
CTX-Quality-Control – Deployment Plan
CTX-Quality-Control.studiopkg

Abbreviations used in this Document

SQL Structured Query Language

Requirements

The CTX-Quality-Control module requires the following:

- SQL Server 2016 or newer
- Minimum Cortex v6.5 installed on the Cortex Application Server
- Windows PowerShell v5 or later

Integration

Integration with Third-Party Systems

None Required.

Integrating with Existing Infrastructure

None Required.

1 Overview

The quality control module allows users to assess quality metrics for their flows based on the Cortex Best Practice Guidelines. For each flow the following metrics are calculated:

- Description coverage
- Number of time savings blocks
- Number of disconnected blocks
- Number of unused variables

A flow or subtask will be marked with a pass or fail based on the configured threshold for these metrics.

Flow quality can be assessed from a LivePortal flow (QC-View-Server-Flows-Quality) or through an Excel document that can be generated using the flow QC-Create-Flow-Report

2 Configuration

QualityControlConfig.txt contains the following parameters:

- DatabaseServer: The name of the server with the Cortex databases
- DatabaseName: The Cortex database containing the flow information, this should be CortexWeb
- PowerShellDomain: Domain of the PowerShell user, leave empty string if PowerShell block authentication isn't required
- PowerShellUser: PowerShell username, leave empty string if PowerShell block authentication isn't required
- PowerShellUser: PowerShell user password, leave empty string if PowerShell block authentication isn't required
- QualityControlPowerShellModulesPath: Path to quality control modules
- CortexRepositoriesPath: Path to cortex web repository
- DisconnectedBlocksThreshold: Threshold maximum number of disconnected blocks
- UnusedVariablesThreshold: Threshold maximum number of unused variables
- DescriptionCoverageThreshold: Threshold for the percentage of blocks that have descriptions
- TimeSavingsThreshold: Threshold for minimum number of time saving blocks
- StateErrorHandlersThreshold: Threshold for maximum number of state error handler blocks
- MailServer: Name of the mail server, required if you intend to send emails
- MailServerPort: Port of the mail server, required if you intend to send emails
- MailServer: Email address of the report sender, required if you intend to send emails

3 UI Guide

The UI will allow you to select a number of flows to be analysed and it will display the result of the analysis.

Start the UI interface by executing the service request named QC-View-Server-Flows-Quality within LivePortal. A list as shown in Figure 1 will be displayed. You can select a group of flows/subtasks, please note that nested groups are displayed with a '\'. All flows within a selected group will be tested, including flows contained within child groups.

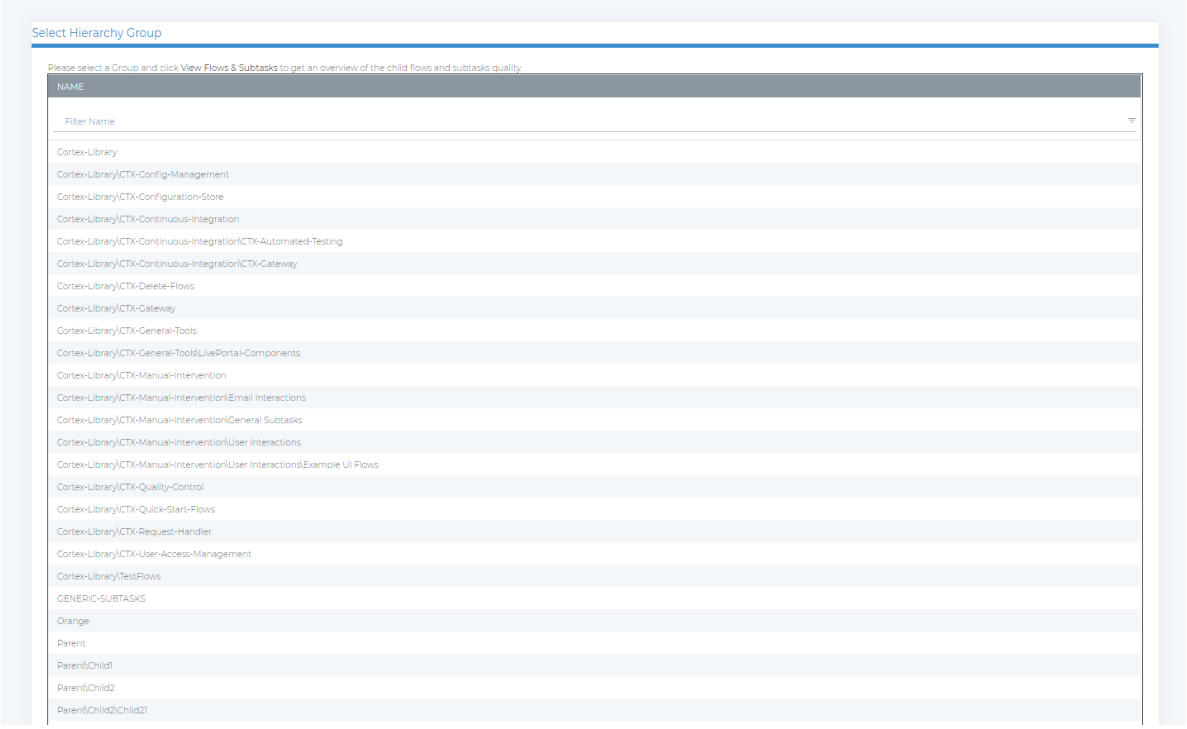


Figure 1 - List of flow groups, flows and subtasks that can be selected for analysis

Once a group is selected the checks will begin to run, this may take a few minutes if lots of flows need to be processed. When the analysis ends, a page will be shown with an overview of the flows analysed as illustrated in Figure 2. You can select a flow or subtask and click View Report to see more details of their compliance with the quality standards.

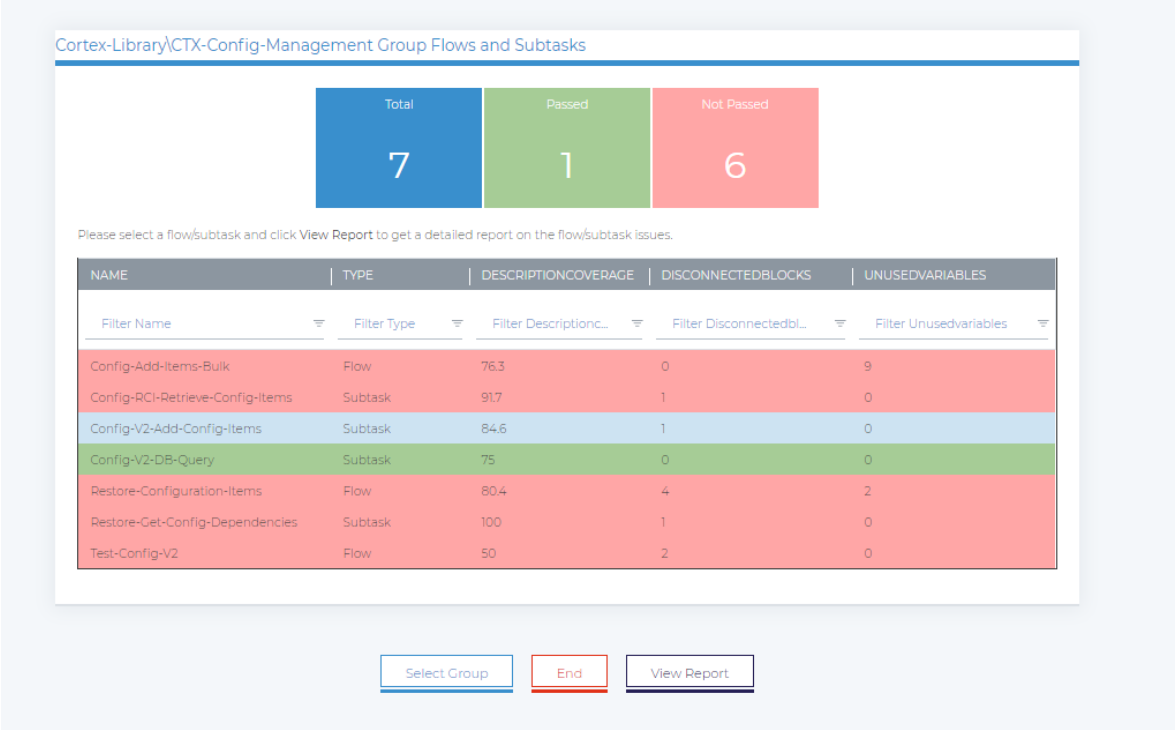


Figure 2 - Result of the flows and subtasks analysis

The report view provides more specific information on disconnected blocks and unused variables as illustrated in Figure 3

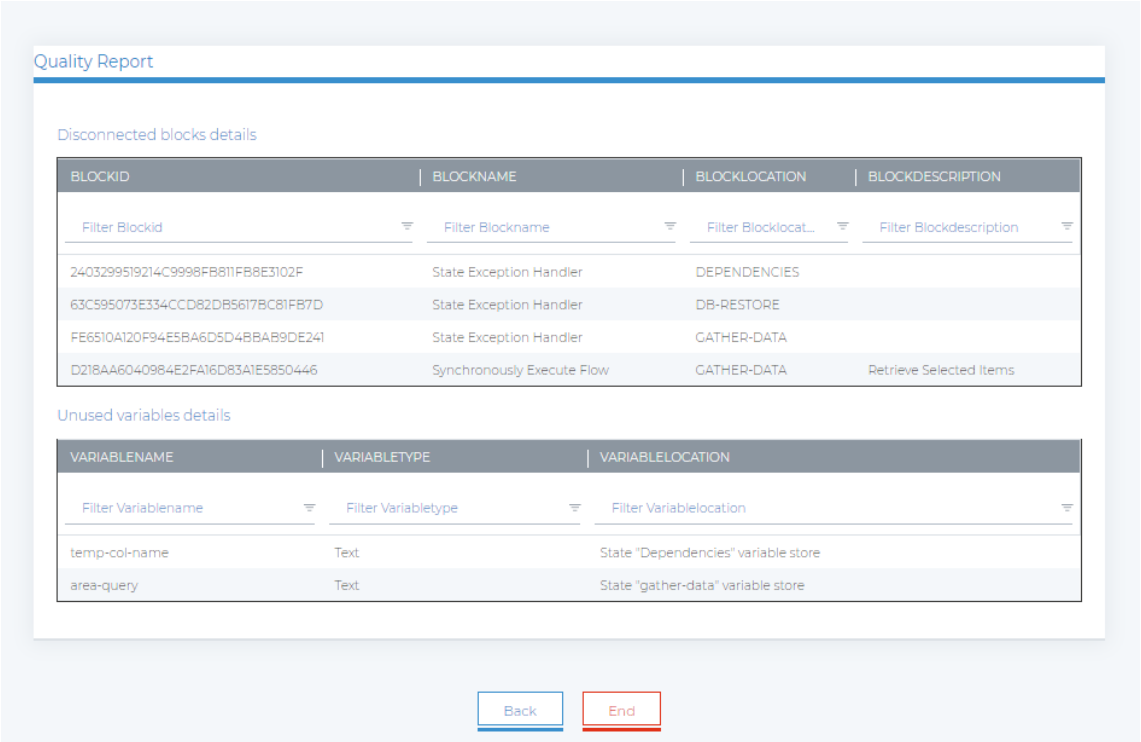


Figure 3 - Flow/subtask quality report

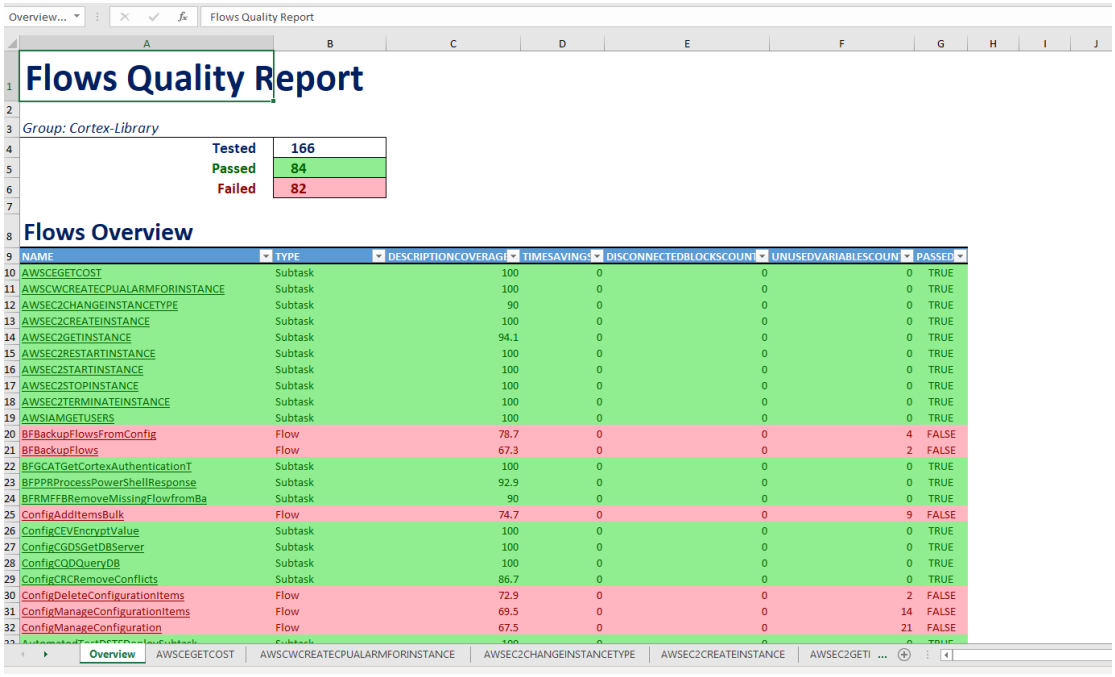
4 Excel Report Generator

This is the recommend method for viewing reports on large numbers of flows. The flow (QC-Create-Flow-Report) takes 3 inputs:

- i_Group: The group of flows/subtasks that you wish to evaluate. Nested groups should take the format: 'Cortex-Library\CTX-Quality-Control'
- i_Contacts: (Optional) CSV of emails to send the report to
- i_FileWritePath: The folder that you wish to save the report to

It is normally useful to create a scheduled flow that generates a report on a periodic basis. Cortex provides a UI to schedule flows as a GitHub module (CTX-Task-Scheduler) that can be found **here**.

The flow will create an Excel document, the first tab will show an Overview as displayed in Figure 4, click on a flow/subtask name will show a detailed view



Flows Quality Report

Group: Cortex-Library

Tested	166
Passed	84
Failed	82

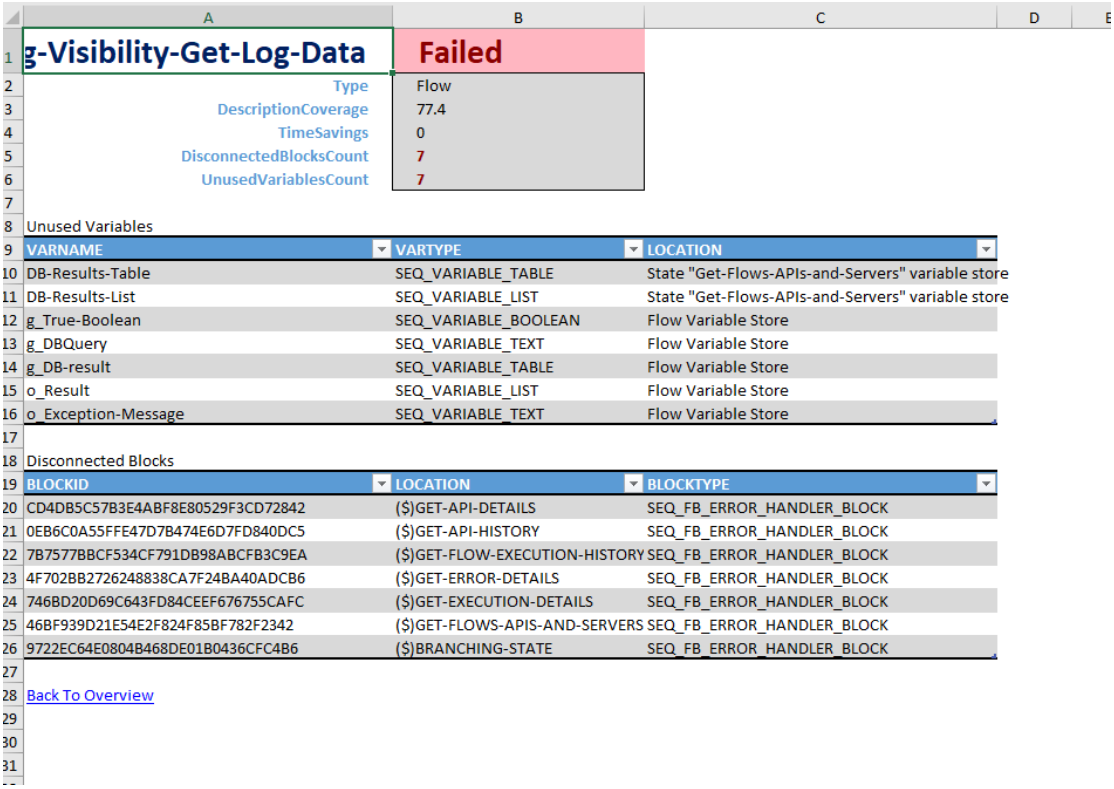
Flows Overview

NAME	TYPE	DESCRIPTIONCOVERAGE	TIMESAVINGS	DISCONNECTEDBLOCKSCOUNT	UNUSEDVARIABLESCOUN	PASSED
AWSCEGETCOST	Subtask	100	0	0	0	TRUE
AWSCWCREATECPUALARMFORINSTANCE	Subtask	100	0	0	0	TRUE
AWSEC2CHANGEINSTANCETYPE	Subtask	90	0	0	0	TRUE
AWSEC2CREATEINSTANCE	Subtask	100	0	0	0	TRUE
AWSEC2GETINSTANCE	Subtask	94.1	0	0	0	TRUE
AWSEC2RESTARTINSTANCE	Subtask	100	0	0	0	TRUE
AWSEC2STARTINSTANCE	Subtask	100	0	0	0	TRUE
AWSEC2STOPINSTANCE	Subtask	100	0	0	0	TRUE
AWSEC2TERMINATEINSTANCE	Subtask	100	0	0	0	TRUE
AWSIAMGETUSERS	Subtask	100	0	0	0	TRUE
BFBackupFlowsFromConfig	Flow	78.7	0	0	4	FALSE
BFBackupFlows	Flow	67.3	0	0	2	FALSE
BFGCATGetCortexAuthenticationT	Subtask	100	0	0	0	TRUE
BFPPRProcessPowerShellResponse	Subtask	92.9	0	0	0	TRUE
BFRMFFBRemoveMissingFlowfromBa	Subtask	90	0	0	0	TRUE
ConfigAddItemsBulk	Flow	74.7	0	0	9	FALSE
ConfigCEVEncryptValue	Subtask	100	0	0	0	TRUE
ConfigCGDSGetDBServer	Subtask	100	0	0	0	TRUE
ConfigCQDQueryDB	Subtask	100	0	0	0	TRUE
ConfigCRCRemoveConflicts	Subtask	86.7	0	0	0	TRUE
ConfigDeleteConfigurationItems	Flow	72.9	0	0	2	FALSE
ConfigManageConfigurationItems	Flow	69.5	0	0	14	FALSE
ConfigManageConfiguration	Flow	67.5	0	0	21	FALSE

Overview | AWSCEGETCOST | AWSCWCREATECPUALARMFORINSTANCE | AWSEC2CHANGEINSTANCETYPE | AWSEC2CREATEINSTANCE | AWSEC2GETI ...

Figure 4 - Flow and subtask analysis excel report overview tab

In Figure 5 you can see a detailed view of a specfic flow or subtask, it gives more information on the unused variables and disconnected blocks. The ‘Back to Overview’ link at the bottom of the tab will take you back to the overview page.



	A	B	C	D
1	g-Visibility-Get-Log-Data	Failed		
2	Type	Flow		
3	DescriptionCoverage	77.4		
4	TimeSavings	0		
5	DisconnectedBlocksCount	7		
6	UnusedVariablesCount	7		
7				
8	Unused Variables			
9	VARNAME	VARTYPE	LOCATION	
10	DB-Results-Table	SEQ_VARIABLE_TABLE	State "Get-Flows-APIs-and-Servers" variable store	
11	DB-Results-List	SEQ_VARIABLE_LIST	State "Get-Flows-APIs-and-Servers" variable store	
12	g_True-Boolean	SEQ_VARIABLE_BOOLEAN	Flow Variable Store	
13	g_DBQuery	SEQ_VARIABLE_TEXT	Flow Variable Store	
14	g_DB-result	SEQ_VARIABLE_TABLE	Flow Variable Store	
15	o_Result	SEQ_VARIABLE_LIST	Flow Variable Store	
16	o_Exception-Message	SEQ_VARIABLE_TEXT	Flow Variable Store	
17				
18	Disconnected Blocks			
19	BLOCKID	LOCATION	BLOCKTYPE	
20	CD4DB5C57B3E4ABF8E80529F3CD72842	($)GET-API-DETAILS	SEQ_FB_ERROR_HANDLER_BLOCK	
21	0EB6C0A55FFE47D7B474E6D7FD840DC5	($)GET-API-HISTORY	SEQ_FB_ERROR_HANDLER_BLOCK	
22	7B7577BBCF534CF791DB98ABCFB3C9EA	($)GET-FLOW-EXECUTION-HISTORY	SEQ_FB_ERROR_HANDLER_BLOCK	
23	4F702BB2726248838CA7F24BA40ADCB6	($)GET-ERROR-DETAILS	SEQ_FB_ERROR_HANDLER_BLOCK	
24	746BD20D69C643FD84CEEF676755CAFC	($)GET-EXECUTION-DETAILS	SEQ_FB_ERROR_HANDLER_BLOCK	
25	46BF939D21E54E2F824F85BF782F2342	($)GET-FLOWS-APIS-AND-SERVERS	SEQ_FB_ERROR_HANDLER_BLOCK	
26	9722EC64E0804B468DE01B0436CFC4B6	($)BRANCHING-STATE	SEQ_FB_ERROR_HANDLER_BLOCK	
27				
28	Back To Overview			
29				
30				
31				

Figure 5 - Flow and subtask analysis excel report detailed report tab

5 Flows

5.1 QC-Create-Flow-Report

5.1.1 Overview

Creates an excel report, writes it to a file and emails it based on a given flow group

5.1.2 Input variables

Name	Type	Comments
i_Group	Text	The group of flows/subtasks that you wish to evaluate. Nested groups should take the format: 'Cortex-Library\CTX-Quality-Control'
i_Contacts	Text	(Optional) CSV of emails to send the report to
I_FileWritePath	Text	The folder that you wish to save the report to

5.2 QC-Get-Group-Quality

5.2.1 Overview

Evaluates flow quality for a given group

5.2.2 Input variables

Name	Type	Comments
i_Group	Text	The group of flows/subtasks that you wish to evaluate. Nested groups should take the format: 'Cortex-Library\CTX-Quality-Control'

5.2.3 Output variables

Name	Type	Comments
o_Group-Quality	Structure	Structured data for group quality

5.3 QC-View-Server-Flows-Quality

5.3.1 Overview

LivePortal flow to display flow quality for a given group

5.3.2 Input variables

None

5.3.3 Output variables

None

6 Troubleshooting

6.1 Maximum message size

Failed to communicate with Cortex Powershell agent at localhost:22100. The following error occured:

System.ServiceModel.CommunicationException: The maximum message size quota for incoming messages (65536) has been exceeded.

This means that too much data is being passed through the PowerShell interface, you may experience this running large groups of flows. The maximum message size can be increased with the following steps:

1. Open C:\Program Files (x86)\Cortex\Cortex PowerShell Agent Service\Innovise.Cortex.Server.Api.PowerShell.Agent.exe.config
2. Under <services> find the <endpoint> with binding=”netTcpBinding”. Add name="netTcpBinding" to the tag and change the value of bindingConfiguration to "netTcpBinding". The opening <endpoint> tag should look similar to this:

```
<endpoint name="netTcpBinding"
        address="net.tcp://localhost:22101/CortexPowerShellService"
        binding="netTcpBinding"
        bindingConfiguration="netTcpBinding"
        contract="Innovise.Cortex.Server.Api.PowerShell.Core.IPowerShellAgentService">
```

3. Open C:\Program Files (x86)\Cortex\Cortex Generic Interface Service\Innovise.Cortex.Server.Host.exe.config
4. Between </startup> and <applicationSettings> insert the following:

```
<system.serviceModel>
  <bindings>
    <netTcpBinding>
      <binding name="netTcpBinding" closeTimeout="00:01:00" openTimeout="00:01:00"
receiveTimeout="00:59:00" sendTimeout="00:01:00"
hostNameComparisonMode="StrongWildcard" listenBacklog="10"
maxBufferPoolSize="100000000" maxBufferSize="100000000" maxConnections="10"
maxReceivedMessageSize="100000000">
        <readerQuotas maxDepth="32" maxStringContentLength="100000000"
maxArrayLength="100000000" maxBytesPerRead="100000000"
maxNameTableCharCount="100000000" />
      </binding>
    </netTcpBinding>
        </bindings>
</system.serviceModel>
```

5. Go to services and restart the Cortex Powershell Agent service and the Cortex Generic Interface service.

6.2 Timeout

Failed to communicate with Cortex Powershell agent at localhost:22100. The following error occured: System.ServiceModel.CommunicationException: The socket connection was aborted. This could be caused by an error processing your message or a receive timeout being exceeded by the remote host, or an underlying network resource issue

It is likely that the receive timeout on the Cortex PowerShell agent is too short. The receive timeout can be increased with the following steps:

1. Open C:\Program Files (x86)\Cortex\Cortex PowerShell Agent Service\Innovise.Cortex.Server.Api.PowerShell.Agent.exe.config
2. Find the recieveTimout attribute on the binding named netTcpBinding. Change the receiveTimeout value from its default value of 00:10:00 to 00:59:00 and save the file
3. Restart the Cortex PowerShell agent service

