ControlPoint

Software Version 5.7.0

Best Practices Guide

Document Release Date: July 2019
Software Release Date: July 2019

Legal notices

Copyright notice

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

You can check for more recent versions of a document through the MySupport portal. Many areas of the portal, including the one for documentation, require you to sign in with a Software Passport. If you need a Passport, you can create one when prompted to sign in.

Additionally, if you subscribe to the appropriate product support service, you will receive new or updated editions of documentation. Contact your Micro Focus sales representative for details.

Support

Visit the MySupport portal to access contact information and details about the products, services, and support that Micro Focus offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- Search for knowledge documents of interest
- Access product documentation
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts
- Submit and track service requests
- Contact customer support
- View information about all services that Support offers

Many areas of the portal require you to sign in with a Software Passport. If you need a Passport, you can create one when prompted to sign in. To learn about the different access levels the portal uses, see the Access Levels descriptions.

Contents

Chapter 1: Best practices for SQL Server

This chapter provides the following tasks you can perform to improve SQL Server performance:

- Configure Windows power options on the server
- Configure SQL Server memory options
- Configure indexes and statistics

Configure Windows power options on the server

In Windows, the default power settings balance power efficiency and performance. For SQL Server to have consistent, predictable, and high performance, set the power option to High Performance. This additional processing capacity comes with higher power utilization.

> **TIP:**
> You can configure this setting on new or existing installations of ControlPoint.

To set the Windows power plan

1. Open **Control Panel** > **Power Option**.
2. Click **High Performance**, and then click **OK**.

   The server power options are set.

Configure SQL Server memory options

SQL Server Memory Manager uses two server memory options, **min server memory** and **max server memory** to manage the amount of memory allocated to a SQL Server process.

After installing SQL Sever, set the maximum amount of server memory instead of using the default value. For example, reserve 1 GB of RAM for the OS, 1 GB for each 4 GB of RAM installed from 4 to 16 GB, and then 1 GB for every 8 GB RAM installed over 16 GB RAM.

For information about how SQL Server uses these options to allocate memory and the default value for each option, see your SQL Server documentation.

> **TIP:**
> You can configure this memory option on either new or existing installations of ControlPoint.

Configure indexes and statistics

You should reconfigure database indexes on a weekly basis. Any index which has 30% or more fragmentation should be rebuilt.

Statistics should be updated daily and have Auto Update statistics enabled. For highly active servers with constant updates and inserts it might be beneficial to update stats every hour.

Chapter 2: Best practices for ControlPoint

This section describes the best practices for your ControlPoint environment.

Databases

This section describes the best practices for your ControlPoint databases.

Back up the ControlPoint and IDOL databases

Before you back up the ControlPoint and IDOL databases, prepare the environment by disabling scheduled tasks and stopping services. This ensures that the ControlPoint and IDOL database backups remain in sync.

To prepare the environment

1. Allow any executing policy phases to complete.

   > **NOTE:**
   > Ensure all items in the existing policies are in the `executed` or `failed` status.

2. In the ControlPoint Administration dashboard, disable the Assign Policies and Execute Policies tasks using the Scheduled Tasks, to prevent new policies from being assigned to documents.

   > **NOTE:**
   > Be sure to disable all of the tasks: Normal, Low and High priority.

3. Check the Distributed Connector queue by issuing the command:

   `http://`
   `distributedconnectorhost``:`*`port`*`/a=queueinfo&queuename=fetch&queueaction=getstatus`

   If the Distributed Connector is working with HTTPS, check the queue by issuing the command:

   `https`
   `://distributedconnectorhost`
   `:port``/a=queueinfo&queuename=fetch&queueaction=getstatus`

   The default port number is 7000.

   All actions should be `Finished`.

4. When all connector actions and executing policy phases have completed, stop the following services:

   a. ControlPoint Engines

   b. Distributed Connector

c. Individual connectors and Connector Framework Services.

The services are stopped.

5. Back up the ControlPoint databases.
   - ControlPoint
   - ControlPoint Audit
   - ControlPointMetaStore
   - ControlPointMetaStore Tags
   - ControlPoint Document Tracking
   - ReportServer. Available if your environment is configured for reports.
   - ReportServerTempDB. Available if your environment is configured for reports
6. Back up the IDOL databases.

Compact stored procedure

When a Compact stored procedure job does not complete before the next scheduled run, then both instances of Compact will run. This slows down the database performance and may prevent ingestion and other operations from running.

Scenario

The Compact stored procedure runs once a week, and its purpose is two-fold:

- To delete any deleted repositories and their documents and document-related information which exist in several ControlPoint tables.
- To remove unused hashes for deleted documents as a result of incremental scans or policy executions.

Solution

The following modifications have been made to the Compact stored procedure:

- Prevent more than one Compact job from running at a time.
- Always delete all repositories that are marked for deletion.
- Perform the cleanup of unused hashes on a limited number of repositories.

Two new settings have been introduced to the **ControlPointMetaStore.Metadata.Settings** table to control the Compact stored procedure. You can adjust the settings for your particular ControlPoint environment.

Setting name	Description
Compact NoIngestTimeMins	The number of minutes of no ingestion activity to

Setting name	Description
	wait before unused hash cleanup runs. **Default:** 15 **NOTE:** This setting was hard-coded in previous releases.
CompactNumReposToCleanupUnusedHash	The maximum number of repositories to perform the cleanup of unused hashed cleanup on. **Default:** -1 (all repositories) **NOTE:** This setting was hard-coded in previous releases.

If you feel the Compact stored procedure is stuck and not completing after one week, you can clear the IsRunning flag.

To clear the flag, run the following SQL command

```
UPDATE [ControlPointMetaStore].[Metadata].[CompactLock] set IsRunning = 0
```

IMPORTANT:
Use caution when deciding to clear the **IsRunning** flag. Ensure that you have waited long enough for the Compact operation to complete.

If you find that the Compact job is taking longer than several days to complete and is affecting the operation of your ControlPoint environment, adjust the Compact stored procedure settings.

If you find that the Compact job is taking longer than several days to complete and is affecting the operation of your ControlPoint environment, adjust the Compact stored procedure settings.

To adjust the Compact stored procedure settings

- Set the **CompactNumReposToCleanupUnusedHash** to 25 percent of the number of repositories.

Example

For 100 repositories, set the CompactNumReposToCleanupUnusedHash to 25.

```
update [ControlPointMetaStore].[MetaStore].[Setting] SET Value=5
where name='CompactNumReposToCleanupUnusedHash'
```

Connectors

This section describes the best practices for various ControlPoint connector components.

Enforce connector security

By default, all users in ControlPoint are able to view the metadata of all items, regardless of IDOL security permissions.

The `SecureMetaStoreContent` setting in `Dashboard\Web.config` controls the view and download options, depending on the IDOL security.

To enforce security

1. Navigate to the following location:

   ```
   \Program Files\Micro Focus\ControlPoint\Dashboard\web.config
   ```

2. Locate the `<appSettings>` section.
3. Edit the `"SecureMetaStoreContent"` value from `"false"` to `"true"`.

   Example

   ```
   <appSettings>
     <add key="SecureMetaStoreContent" value="true"/>
   </appSettings>
   ```

4. Save the file.

Managing the Index batch size

The documents ingested by a Connector from the source repository are processed by a Connector Framework service that then forwards them in batches to the ControlPoint MetaStore service.

The metadata associated with each document varies considerably depending on, for example, whether eduction grammars have been selected for the source repository and how many educed fields are discovered within each document. If the total size of data in each batch of documents the Connector Framework service sends to the ControlPoint MetaStore service is very large, it can affect the CPU and memory usage of both services.

To prevent the Connector Framework and ControlPoint MetaStore services from using an excessively high amount of CPU and memory when you know in advance that document batches are likely to be large, decrease the batch size. For example, if it is known in advance that eduction grammars will be specified that will likely generate a lot of metadata for each document then you should decrease the batch size. To do so, modify the `IndexBatchSize` setting in the `[Indexing]` section of the Connector Framework service configuration file. This setting controls the number of documents per batch. For example, the following configures a maximum batch size of 10 documents per batch:

```
[Indexing]
IndexBatchSize=10
```

Configuration changes and service restarts

For environmental changes to take effect immediately, you must stop and restart the following services.

Changed configuration area	Restart the services
Distributed Connector	ControlPoint Distributed Connector
Edge Filesystem Connector	ControlPoint Edge Filesystem Connector ControlPoint Edge Filesystem Connector Framework ControlPoint Edge Archive
Edge Archive Service	**Windows:** ControlPoint Edge Archive Service **Linux:** `mfloggedfs` process
Filesystem Connector	ControlPoint Filesystem Connector ControlPoint Filesystem Connector Framework
IDOL	ControlPoint IDOL

Scheduled tasks and schedules

ControlPoint includes a number of scheduled tasks to automatically perform jobs that are required to manage policies, generate statistical information for monitoring purposes, and so on. You can control how often these automated tasks run through schedules.

For more information on configuring scheduled tasks and schedules, see the *ControlPoint Administration Guide* or the Administration Help system.

Configure ControlPoint schedules for large systems

The following section describes ControlPoint schedule configurations to use in large ControlPoint systems. Depending on your requirements and hardware, you can combine the solutions in this section as required.

Change the number of scheduler threads

Each ControlPoint Scheduler runs a defined number of threads, each processing a batch of items every time it runs. The default number of threads is eight. The optimal number of threads depends on your requirements and the system processor.

To change the number of Scheduler threads

1. Open the **ControlPoint Configuration Manager**.
2. Click **Engine**.

   The Engine Setting page opens.
3. Under **Engine Settings**, enter the number of threads in the **Enter the number of threads to use to process items** box.

   Micro Focus recommends one thread per core.
4. Click **Deploy**.

   ControlPoint redeploys.

Install multiple ControlPoint schedulers

For high processing volumes, you can install multiple ControlPoint Schedulers on several machines. You must modify the configuration of each Scheduler to point to the ControlPoint SQL Server database and the IDOL server.

Chapter 3: Best practices for policy execution

Policy execution is a crucial functionality in ControlPoint that enables you to apply various actions on documents.

The following chapter describes some tips to ensure the policy execution engine runs properly.

Design of policy execution engine

The policy execution engine is designed to automatically process documents and report results without much user intervention.

Multiple scheduled tasks run in predefined frequencies to ensure that the policy executions run smoothly.

> **NOTE:**
> Disabling or changing the frequency of the scheduled tasks may have undesirable results.

There are some items that depend on user actions:

- **Issues**. Navigate to the Issues Management page in the Administration Dashboard to review issues.

  Choose **Abort** or **Retry** for any failed executions for documents.

- **Conflict Resolutions**. Navigate to the Conflict Management page in the Administration Dashboard to decide which policy should execute when two or more policies are applied to a policy at the same time before any of the policies are executed. For more information, see Monitor policy conflicts, on the next page.

Policy execution

- Allow any executing policy phases to complete before changing the ControlPoint environment.

  > **NOTE:**
  > Ensure all items in the existing policies are in the `executed` or `failed` status, before the changing the environment.

  Database changes, restores of the IDOL content databases or software upgrades that occur while policy executions are running may leave the environment in an inconsistent state.

- When manually applying a category to a policy, select the child level category.

Policy execution logs

Policy execution logs can be found at the following location:

```
\Program Files\Micro Focus\ControlPoint\Engine\Scheduler\Logs
```

For more information, see the Troubleshooting chapter in *ControlPoint Installation Guide*, or the Console Help Center.

Monitor policy conflicts

The number of policy conflicts in ControlPoint can affect the performance of the policy executions.

To optimize policy execution performance, Micro Focus recommends that you keep the number of policy conflicts as low as possible.

> **NOTE:**
> For more information on policy conflicts and how to resolve them, see the *ControlPoint Administration Guide* or Help Center.

To monitor the policy conflicts

1. In the ControlPoint Administration dashboard, navigate to **Administration**, then click **Conflict Management.**

   The Conflict Management page opens, listing any policy conflicts.

   You can manually resolve the conflicts, or configure ControlPoint to automatically attempt to resolve them.

Policy execution global settings

You can configure the system by using settings stored centrally in the **ControlPoint.dbo.CPGlobalSettings** table in the ControlPoint database.

Consider the following when changing settings in the Global Settings table in the ControlPoint database:

- In SQL Server, back up the ControlPoint database before attempting to make any changes to the **CPGlobalSettings** table.
- Empty values for items in the Global Setting table do not imply that the setting is zero (0).

  Micro Focus recommends that you retain the default value of a global setting instead of setting a value to zero.
- Care should be taken when changing values in the Global Settings table.

  Erroneous values in the Global Settings table can lead to lower performance or blocked execution

progress in your ControlPoint environment.

- For the default setting of each item, see the **DefaultValue** column in the Global Settings table.

SettingName	Description
Autonomy.ControlPoint.IdolDocument Processing BatchSizeWithResults	This setting affects many areas of ControlPoint, such as how many documents are discovered in a single discoverer action. It also affects the ControlPoint Engine queries IDOL for documents. For policy executions, this setting defines the batch size that determines how many items are sent to the connector as a batch. As a result, it determines the batch size of callbacks received from the connectors. When this value is set too high, it can cause IDOL to become unresponsive. **NOTE:** Setting this value to a larger value may accelerate policies, but it is not recommended.
Autonomy.ControlPoint.ExecutionLog ProcessingBatchSize	The batch size that determines how many ExecutionLog items a policy engine thread can put a lock on. For example if the value is set to 1000, each engine thread can put a lock on 1000 ExecutionLog items. If there are 4 engine threads, the engine can process 4000 items simultaneously. Setting the value too high causes a single engine thread to put a lock on a large amount of ExecutionLog items, eliminating the benefits of concurrent processing in the policy engine. **NOTE:** • This value must be bigger or equal to the value in Autonomy.ControlPoint.IdolDocumentProcessingBatchSizeWithResults. Otherwise the engine will fail to put a lock on any ExecutionLog items. • This value must be smaller than the Autonomy.ControlPoint.CallbackQueueLimit, preferably one-tenth of it. If this value is too close to Autonomy.ControlPoint.CallbackQueueLimit, it can lead to poor performance of the engine.
Autonomy.ControlPoint.CallbackQueu	The limit that determines how many document action

SettingName	Description
eLimit	requests can be sent to a connector group at a time. For example if the value is set to 10000, the engine will send at most 10000 document action requests to all File System connectors, and at most 10000 document action requests to all Exchange connectors. • Setting this value too high may overload the connector and cause the connector to be non-responsive. • Setting this value too low may result in poor performance in the policy engine. **NOTE:** If this value is smaller than ExecutionLogProcessingBatchSize, the Sender phase of the policy execution engine will fail to send any document action request to connectors due to mechanisms to avoid overloading the connectors.
Autonomy.ControlPoint.ExecutionLogLimit	The limit that determines the maximum size of the ExecutionLog table. The discovery phase of policy executions will put ExecutionLog entries in the ExecutionLog table. Once the limit is reached, the discovery phase will pause. • Setting this limit too low may cause the policy executions to slow down. • Setting this value too high may cause the ExecutionLog items to timeout as they wait to be get executed by the other policy execution phases.
Autonomy.ControlPoint.ExecutionLogExpireMinutes	Queue Verifier is a phase in policy execution schedule that queries the connector for ExecutionLog items that has not heard back from the connector for a period of time in case of lost callbacks. The value defines that period for time here.
Autonomy.ControlPoint.IndexBatchSize	This should be set to the same value as `[Indexing]IndexBatchSize` in Connector Framework configuration files. It is used to control the number of orphaned items deleted at a time from IDOL and MetaStore.
Autonomy.ControlPoint.Security.CacheTimeOut	Timeout setup for security related cache. Determines how often to refresh the domain cache.

Chapter 4: Best practices for ControlPoint security

This section describes security-related best practices for your ControlPoint environment.

Encrypt ControlPoint configuration files

Microsoft .Net 2.0 contains a feature called Protected Configuration that enables you to encrypt different sections of configuration files. You can use this feature to encrypt sections of ControlPoint configuration files that contain sensitive information.

This section describes how to encrypt sensitive data in the `Dashboard.config` file. You can use this same process to encrypt sensitive information in other configuration files.

To encrypt data in a configuration file

1. Create an RSA Container.

   Create a container for the encryption keys using the `aspnet_regiis` tool, typically located in `WINDOWS\Microsoft.Net\Framework\v2.0.*`, and provide it a name, such as `ControlPointKeys`:

   ```
   aspnet_regiis -pc "ControlPointKeys" –exp
   ```

2. Grant access to the RSA Container.

   Grant the user that the IIS pool uses access to the new RSA Container. In the following example, the user is: `IIS APPPOOL\ControlPointAppPool40`.

   ```
   aspnet_regiis -pa "ControlPointKeys" "IIS APPPOOL\ControlPointAppPool40"
   ```

3. Specify a Protected Configuration Provider.

   In the `dashboard.config` file, use a text editor to do the following:

   a. Locate the `<configuration>` section, and then add a `<configProtectedData>` section as a child of it.

   b. In the new child section, add an instance of the `RSAProtectedConfigurationProvider` class named `"ControlPointProvider"` that uses the machine-level RSA key container named `"ControlPointKeys"`.

   For example:

   ```
   <configProtectedData>
       <providers>
          <add name="ControlPointProvider"
               type="System.Configuration.RsaProtectedConfigurationProvider,
                     System.Configuration, Version=2.0.0.0,
   ```

```
                    Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a,
                    processorArchitecture=MSIL"
                keyContainerName="ControlPointKeys"
                useMachineContainer="true" />
        </providers>
    </configProtectedData>
```

The `dashboard.config` file now has the following content:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="idolServersConfigurationSection" type="Autonomy.Idol.SystemConfigurationSection.IdolServersConfig
  </configSections>
  <configProtectedData>
    <providers>
      <add name="ControlPointProvider"
          type="System.Configuration.RsaProtectedConfigurationProvider, System.Configuration, Version=2.0.0.0,
              Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a,
              processorArchitecture=MSIL"
          keyContainerName="ControlPointKeys"
          useMachineContainer="true" />
    </providers>
  </configProtectedData>
  <connectionStrings>
    <remove name="LocalSqlServer" />
    <add name="Entities" connectionString="metadata=res://*/Entities.DashboardModel.csdl|res://*/Entities.Dashboard
    <add name="ControlPointEntities" connectionString="metadata=res://Autonomy.ControlPoint.DataAccess.Providers.Ent
    <add name="ControlPointAuditEntities" connectionString="metadata=res://Autonomy.ControlPoint.Audit/Entities.Cont
    <add name="ControlPointTrackingEntities" connectionString="metadata=res://*/Entities.ControlPointTracking.csdl|r
  </connectionStrings>
  <idolServersConfigurationSection default="Default System">
    <idolServers>
      <idolServer name="Default Server" dihHost="16.73.15.114" dihPort="9070" dihIndexPort="9071" connectorHost="16.
    </idolServers>
  </idolServersConfigurationSection>
</configuration>
```

4. Rename the `dashboard.config` file for use with the `aspnet_regiis` tool.

   The `aspnet_regiis` tool works only with files named "`web.config`"; however, the data to encrypt resides in the `dashboard.config` file. Therefore, do the following:

   a. Rename the existing `web.config` file to temporarily save it under a different name.

      For example:

      ```
      rename web.config web.tmp.config
      ```

   b. Rename the `Dashboard.config` file to `web.config`.

      ```
      rename dashboard.config web.config
      ```

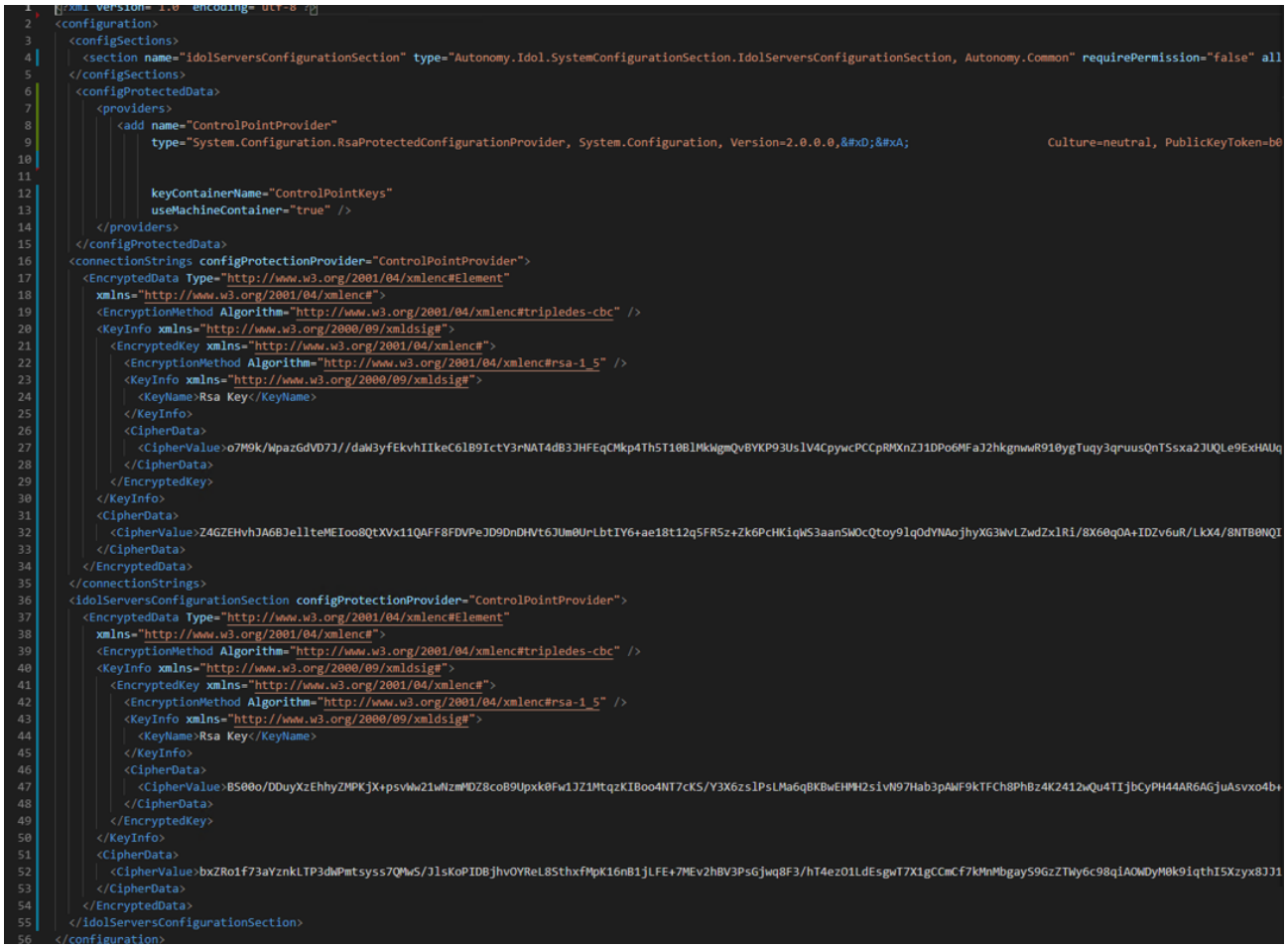
5. Encrypt sections of the new `web.config` file.

   Encrypt the `ConnectionStrings` and `idolServersConfigurationSection` sections of the `web.config` file using the following commands. Each command performs the encryption using the information in the `ControlPointProvider` section and the application name from IIS.

   - `aspnet_regiis -pe "connectionStrings" -app "/ControlPoint" -prov "ControlPointProvider"`

- `aspnet_regiis -pe "idolServersConfigurationSection" -app "/ControlPoint" -prov "ControlPointProvider"`

After running these commands, the `web.config` file is encrypted:



6. Rename the configuration files back to their original names.

   Rename the `web.config` file back to its original `dashboard.config` file name and the temporary file back to `web.config`.

   For example:

   - `rename web.config dashboard.config`
   - `rename web.tmp.config web.config`

The sensitive sections of the dashboard.config file are now encrypted and the data will be available to ControlPoint as .Net handles the encryption transparently. For more information, see https://docs.microsoft.com/en-us/previous-versions/2w117ede(v=vs.140).

Chapter 5: Global Settings considerations

You can configure the system by using settings stored centrally in the **ControlPoint.dbo.CPGlobalSettings** table in the ControlPoint database.

Consider the following when changing settings in the Global Settings table in the ControlPoint database:

- In SQL Server, back up the ControlPoint database before attempting to make any changes to the **CPGlobalSettings** table.
- Empty values for items in the Global Setting table do not imply that the setting is zero (0).

  Micro Focus recommends that you retain the default value of a global setting instead of setting a value to zero.
- Care should be taken when changing values in the Global Settings table.

  Erroneous values in the Global Settings table can lead to lower performance or blocked execution progress in your ControlPoint environment.
- For the default setting of each item, see the **DefaultValue** column in the Global Settings table.

Selected Global Settings table values

Setting name	Description
Autonomy.ControlPoint.DataAnalysis.TagBatchSize	Used by the Data Analysis service to perform tagging of documents, for example, tagging documents for duplicates. This value specifies the batch size of the tagging in Data Analysis service. This value must be a non-zero positive number. The default is 10000.
Autonomy.ControlPoint.IdolDocumentProcessing BatchSizeWithResults	This setting affects many areas of ControlPoint, such as how many documents are discovered in a single discoverer action. It also affects the ControlPoint Engine queries IDOL for documents. When this value is set too high, it can cause IDOL to become unresponsive. **NOTE:** Setting this value to a larger value may accelerate policies, but it is not recommended.
Autonomy.ControlPoint.ScheduleLock	Some policy phases can place a lock on a schedule in the

Setting name	Description
AgeMinutes	**CPScheduleLock** table. For example, the conflict resolution resolver places a lock on the policy execution schedule when the conflict resolution resolver is busy. This value specifies the timeout period of the lock. **NOTE:** Micro Focus recommends that this value not be set high, as items take a long time to expire.
Autonomy.ControlPoint.ExecutionLog ProcessingBatchSize	The batch size that determines how many ExecutionLog items a policy engine thread can put a lock on. For example, if the value is set to 1000, each engine thread can put a lock on 1000 ExecutionLog items. If there are 4 engine threads, the engine can process 4000 items simultaneously. Setting the value too high causes a single engine thread to put a lock on a large amount of ExecutionLog items, eliminating the benefits of concurrent processing in the policy engine. **NOTE:** • This value must be bigger or equal to the value in Autonomy.ControlPoint.IdolDocumentProcessingBatchSizeWithResults. Otherwise the engine will fail to put a lock on any ExecutionLog items. • This value must be smaller than the Autonomy.ControlPoint.CallbackQueueLimit, preferably one-tenth of it. If this value is too close to Autonomy.ControlPoint.CallbackQueueLimit, it can lead to poor performance of the engine.
Autonomy.ControlPoint.CallbackQueue Limit	The limit that determines how many document action requests can be sent to a connector group at a time. For example if the value is set to 10000, the engine will send at most 10000 document action requests to all File System connectors, and at most 10000 document action requests to all Exchange connectors. • Setting this value too high may overload the connector and cause the connector to be non-responsive. • Setting this value too low may result in poor

Setting name	Description
	performance in the policy engine. If this value is smaller than ExecutionLogProcessingBatchSize, the Sender phase of the policy execution engine will fail to send any document action request to connectors due to mechanisms to avoid overloading the connectors.
Autonomy.ControlPoint.ExecutionLogLimit	The limit that determines the maximum size of the ExecutionLog table. The discovery phase of policy executions will put ExecutionLog entries in the ExecutionLog table. Once the limit is reached, the discovery phase will pause. • Setting this limit too low may cause the policy executions to slow down. • Setting this value too high may cause the ExecutionLog items to timeout as they wait to be get executed by the other policy execution phases.
Autonomy.ControlPoint.ExecutionLogExpireMinutes	Queue Verifier is a phase in policy execution schedule that queries the connector for ExecutionLog items that has not heard back from the connector for a period of time in case of lost callbacks. The value defines that period for time here.
Autonomy.ControlPoint.IndexBatchSize	This should be set to the same value as `[Indexing]IndexBatchSize` in Connector Framework configuration files. It is used to control the number of orphaned items deleted at a time from IDOL and MetaStore.
Autonomy.ControlPoint.Security.CacheTimeOut	Timeout setup for security related cache. Determines how often to refresh the domain cache.

Appendix A: Export statistics

You can use a statistics export utility to export data to Microsoft Excel. The type of data exported depends on the state of the repository.

- Statistics can be exported from any analyzed repository.
- Metrics can be requested from any unanalyzed repository.
- Metrics can be requested from any data set that can be identified using an IDOL query. For example, all documents that have a specific policy applied, all documents authored by a given user, and so on.

Sample Microsoft Excel templates are provided with the utility.

Before you begin

Install Microsoft Excel to the ControlPoint server.

To export statistics

1. Run the Statistics Export Utility, which is available at the following location:

   ```
   ControlPoint x64\ControlPoint Utilities\Statistics Export
   Utility\ControlPointStatisticsUtility.exe
   ```

   The ControlPoint Analysis window opens.

2. Enter the host name in the **Host** box, and then click **OK**.

   The export dialog box appears. The Analysis Tasks section lists all analyzed repositories on the host system.

3. (Optional) To re-analyze a repository, select it, and then click **Re-analyze**.

4. (Optional) **To add a custom analysis task**

   a. Click **New**.

      The New Custom Analysis Task dialog box opens.

   b. Enter a **Task Name**.

   c. Enter or select **IDOL Query Parameters**.

   d. Click **OK**.

      The task is added to the list.

5. Select an analysis task.

6. In the **Export Task** section, select a Microsoft Excel template from the list, and then click **Export**.

   The data exports to Excel and appears according to the selected template. Potential Obsolete and Trivial disk space appears in the Obsolete-AllPotential and Trivial-AllPotential charts.

Statistics Utility command line interface

The ControlPoint Statistics utility now supports a command line interface for exporting results.

Location

```
ControlPoint x64\ControlPoint Utilities\Statistics Export
Utility\ControlPointStatisticsUtility.exe
```

Synopsis

```
ControlPointStatisticsUtility.exe -dahost <hostname> -enablehttps 0|1
-sqlhost <hostname> -authtype 0|1 -dataset repo -action 0|2|3
-templatepath <path> -exportpath <path>
```

Options

Parameter	Required	Description
-dahost	Required	Specify the host name of the Data Analysis service machine.
-sqlhost	Required	Specify the host name of SQL Server machine.
-authtype	Required	Specify the SQL Server authentication type: 0 is Windows user authentication 1 is SQL Server user authentication
-enablehttps	Required	Specify whether the enable HTTPS. 0 is no 1 is yes **NOTE:** Only set to 1 when ControlPoint environment is enabled with HTTPS.
-dataset	Optional	Specify the data set to take action on. Required for export.
-action	Required	Specify the type of action to perform: • 0 is export. • 2 is re-analyze. • 3 is delete.

Parameter	Required	Description
-sqluser	Optional	Specify the user name of a SQL Server user. **NOTE:** Required when `-authtype` is set to 1.
-password	Optional	Specify the password of the SQL Server user. **NOTE:** Required when `-authtype` is set to 1.
-templatepath	Optional	Absolute path of the template file. **NOTE:** Required when the `-action` is set to export (0).
-exportpath	Optional	Absolute path of the export file. **NOTE:** Required when the `-action` is set to export (0).
-taskname	Optional	Name of task to be re-analyzed or deleted. **NOTE:** Required when the `-action` is set to re-analyze (2) or delete (3).

Examples

To export data

```
ControlPointStatisticsUtility.exe -dahost cpserver -enablehttps 0
-sqlhost cpserver -authtype 0 -dataset repo -action 0
-templatepath C:\test\Templates\Blank.xltx -exportpath C:\test\export\repo.xlsx
```

To re-analyze a repository

```
ControlPointStatisticsUtility.exe -dahost cpserver -enablehttps 0
-sqlhost cpserver -authtype 0 -action 2 -taskname myTask
```

To delete a task

```
ControlPointStatisticsUtility.exe -dahost cpserver -enablehttps 0
-sqlhost cpserver -authtype 0 -action 3 -taskname myTask
```

Appendix B: AppSettings in ControlPointTimer.config

The following is a reference for the `<AppSettings>` in `ControlPointTimer.config` file.

Settings	Usage
NumberOfTimerThreads	Number of Threads for the timer engine
ExceptionWaitTime	If 5 exceptions have been thrown in a row wait the amount of time indicated
ClientSettingsProvider.ServiceUri	
SleepSeconds	Thread sleep seconds for ingestion
MaxExecutionFrequencySeconds	Used in Phase execution
CallbackProcessor.MaxInstancesRunning	Used in collect cleanup
CacheExpirationSettingsCSV	CSV for long expiry seconds, short expiry seconds used during ControlpointFrameworkRegistration andPolicyExecutionRegistration
LoadBalancingSettingsCSV	CSV for maxLatestNoWorkCount, maxPhaseIgnoreSeconds, slidingIgnoreSecondsIncrease
ClearLocksAtStartUp	Boolean value. Set this value to true to clear locks in the ExecutionLog table during ControlPoint Scheduler startup. **NOTE:** Set this to true only when ControlPoint has only one Scheduler instance deployed. Engine crashes or unexpected restarts can leave locks on the execution items. Enabling this option clears the locks upon Scheduler start and therefore avoids putting policy executions on hold for a long period of time.
InsertConfigEnabled	Boolean value. Default value is 'false'. Set this value to true to enable querying IDOL and MetaStore for insert configurations.

Settings	Usage
	Setting this value to false allows the engine to skip querying MetaStoreand IDOL for the insert configuration values, thus improving the execution performance for insert actions to target locations. If you need to use declare in place policy phase (for Content Manager target locations only) or custom insert configurations, enable this option. If you do not use insert configurations, setting this option to false will improve performance for policy executions. For more information on Insert Configurations, see the *ControlPoint Administration Guide* or the Help Center.

The following parameters are needed to enable secure connections with IDOL and Connectors

Settings	Usage
SecurePorts	Boolean value, used to determine if the specified metastore port must be added to the metastore port list
MetaStorePort	Port number
LDAPServer	
LDAPBaseObject	
LDAPUseSSL	Boolean to use SSL
LDAPMaxResults	Maximum number of results to retrieve
XMLGroupMembershipFile	Filename containing group information

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Best Practices Guide (Micro Focus ControlPoint 5.7.0)

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to swpdl.controlpoint.docfeedback@microfocus.com.

We appreciate your feedback!