



i-Reporter

オンプレミス版 サーバーストレージ容量の削減方法

2020.05.18
Rev.1.0.0

CIMTOPS

改訂履歴

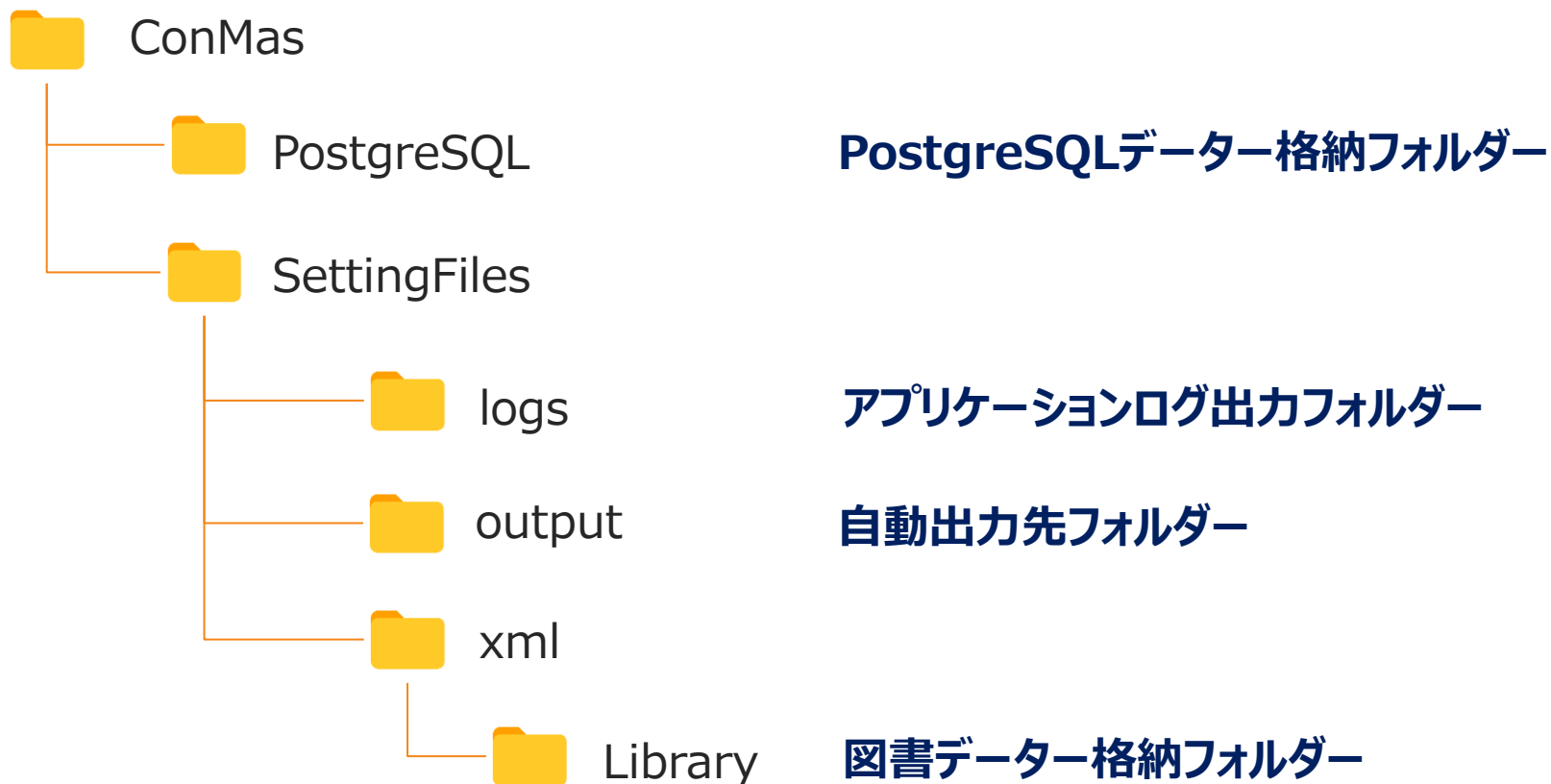
日付	記載ページ	改訂内容
2020-05-18		初版

目次

• 各データの格納場所	P4
• アプリケーションログ出力フォルダー	P5
• 自動出力先フォルダー	P6
• 図書データ格納フォルダー	P7
• PostgreSQLデータ格納フォルダー	P8
• PostgreSQLデータ格納フォルダー容量削減手順	P9
• A) VacuumFullを実行する手順	P10
• B) バックアップ、リストアにてDBを再構築する手順	P11
• 参考資料	
• ログローテーション設定	P13
• VacuumFullバッチサンプル	P14
• DBバックアップバッチサンプル	P16
• DB削除（抹消）バッチサンプル	P17
• DB生成バッチサンプル	P18
• DBリストアバッチサンプル	P19

各データの格納場所

i-Reporterの各データは、以下のフォルダーに格納されています。



アプリケーションログ出力フォルダー

ConMasサーバーサイドプログラムのログ出力フォルダー。
ログファイルのためフォルダーから直接削除が可能です。

【容量の削除方法】

- ・過去のログファイルは、削除します。
- ・ログローテーション設定を利用し、一定の容量以上になると、過去のログファイルから順に削除するように設定します。

※ログローテーション設定手順を参照ください。

自動出力先フォルダー

データ自動出力機能によって出力されるファイルの格納フォルダー。
出力後は、フォルダーから各ファイルを移動、削除することが可能です。

【容量の削除方法】

- ・自動出力機能を利用しない場合

→自動出力機能をOFFにします。

すでに、出力されたファイルが存在する場合は、すべて削除します。

- ・自動出力機能を利用している場合

→連携目的で使用している場合は、連携済みファイルは削除します。

→参照目的で使用している場合は、定期的に別の場所に移動します。

図書データー格納フォルダー

図書データーのファイルが格納されているフォルダー。
図書データーは、**直接フォルダーから削除は出来ません**。

【容量の削除方法】

不要な図書がある場合は、Managerの図書管理より削除します。

“サーバー上”の別ディスクに移動させることで削減が可能です。

※図書管理マニュアルの「ドキュメントの保存場所を変更する」を参照。

PostgreSQLデータ格納フォルダー

PostgreSQLデータベースの全データが格納されているフォルダー。

図書ファイルを除く全てのデータが格納されています。

PostgreSQLは追記型アーキテクチャのDBであることから
レコードの物理削除のみでは利用容量は削減されません。

【容量の削除方法】

容量を削除するには**システム停止を伴う作業が必要**となります。

PostgreSQLデータ格納フォルダーの容量削減手順

手順 1)

i-Reporter帳票を**“物理”削除**します。

物理削除の方法

① ConMasManagerの画面から帳票を選択して物理削除します。

② 外部連携APIオプションを使用して物理削除します。

削減できる容量は上記において物理削除したデータ量となります。

手順 2)

下記の2つの方法のどちらかで、DBの容量を削減します。

運用やデータの容量など状況にあった方のみ、どちらか一つ実施します。

A) PostgreSQLのメンテナンスツールVacuumFullを実行する手順

B) バックアップ、リストアにてDBを再構築する手順

A) VacuumFullを実行する手順

PostgreSQLのメンテナンスツールVacuumFullを利用して容量を削減する方法です。

特徴)

- Vacuum.exeをFullオプション指定して実行するのみで、メンテナンス作業がシンプル。
- 実行検証ができないため、実行にかかる時間が事前に確認できない。
 - ※VacuumFull 実行時間目安を参照
- 実行するには以下条件を満たしている必要がある。
 - 条件：データ格納フォルダーの同一ディスク（パーティション）上にDB使用量の倍の“空き容量”が存在していること。

操作手順)

1. IISを停止します。
2. VacuumFullを実行します。
 - ※バッチサンプル参照
3. IISを開始します。

B) バックアップ、リストアにてDBを再構築する手順

バックアップ、リストアにてデータベースを再構築して容量を削減する方法です。

特徴)

- ・バックアップ、リストアは事前に確認ができるため、メンテナンスに必要な時間を計測することが可能。
- ・ディスクサイズが一杯になっている場合でも、バックアップダンプを別ディスク、または外部ストレージへ出力することで作業が行える。
- ・データベースを一度抹消するため、作業ミスがあるとデータ復元ができなくなる。

操作手順)

1. IISを停止します。
2. データベースを“irepodb”をバックアップします。
※バッチサンプル参照
3. データベース“irepodb”を削除（抹消）します。
※バッチサンプル参照
4. データベース“irepodb”を作成します。
※バッチサンプル参照
5. 2. で取得したバックアップダンプを4. のデータベースにリストアします。
※バッチサンプル参照
6. IISを開始します。

參考資料

ログローテーション設定

設定は、ConMasManager,ConMasWeb（外部連携オプションをご購入いただいているお客様はConMasAPIも）配下のweb.configにて行います。

1. web.configをメモ帳で開きます。
2. "◆◆◆ Log4net設定 ◆◆◆" を検索し移動します。
3. log4netエレメント→appenderエレメント配下に以下を差し込みます。
appenderエレメントは1つのweb.config上に複数箇所あるため、すべてに差し込んでください。

```
***** 差込 *****  
<!-- ファイルサイズで Rollong -->  
<rollingStyle value="Size" />  
<!-- 最大バックアップファイル数 -->  
<maxSizeRollBackups value="10" />  
<!-- ファイルサイズの上限 -->  
<maximumFileSize value="10MB" />  
***** 差込 *****
```

※上記は、1つのログファイルの最大容量を10MBとして、最大で10ファイルバックアップとして過去ログファイルを残す設定となります。この設定は、変更が可能です。

VacuumFullバッチサンプル

```
chcp 65001
SET PGPASSWORD=[PostgreSQLパスワード]
SET VACUUMPGPATH=[vacuumdb.exeのフルパス]
SET DBNAME=[Vacuum対象のデータベース ※既定はirepodb]
SET LOGPATH=[Vacuumログファイルのフルパス]
SET PORT=[PostgreSQLのポート番号 ※既定では5432]

"%VACUUMPGPATH%" -h localhost -U postgres -p %PORT% -f %DBNAME% > %LOGPATH% 2>&1
echo Vacuum ErrorLevel=%ERRORLEVEL% >> %LOGPATH%
```

注意点)

- ・事前にVacuumFull実行条件を確認してください。
- ・実行キャンセルは"Ctrl+C"にて行えます。
- ・実行後、Vacuumログファイルをメモ帳で開き"errorlevel=0"であることを確認します。

VacuumFull 実行時間目安

実行環境情報)

PostgreSQL11

ストレージ：SSD（IOPS上限 = 500, スループット上限（MB/秒） = 60

メモリー：7G

CPU: 2コア

自動帳票作成でDB容量分の帳票を作成し、約5G分の帳票を物理削除して実行
※VacuumFull実行条件を満たした環境にて実行しています。

DB容量(G)	VacuumFull所要時間(分)
10	11
50	106
100	296
200	615

上記は参考値となります。サーバースペック（特にストレージ速度）によって異なります。

DBバックアップバッチサンプル

```
chcp 65001
SET PGPASSWORD=[PostgreSQLパスワード]
SET PGPATH=[pg_dump.exeまでのフルパス]
SET DUMPDB=[ダンプ出力するデータベース名 ※既定ではirepodb]
SET DUMPPATH=[ダンプ出力ファイル名のフルパス]
SET LOGPATH=[ダンプ出力ログファイルのフルパス]
SET PORT=[PostgreSQLのポート番号 ※既定では5432]

"%PGPATH%" -Fc -Z5 -c -h localhost -p %PORT% -U postgres -d %DUMPDB% -f %DUMPPATH% > %LOGPATH%
echo errorlevel=%ERRORLEVEL% >> %LOGPATH%
```

注意点)

- ・実行後、ダンプ出力ログファイルをメモ帳で開き“errorlevel=0”であることを確認します。

DB削除（抹消） バッチサンプル

```
chcp 65001
SET PGPASSWORD=[PostgreSQLパスワード]
SET DROPPGPATH=[dropdb.exeまでのフルパス]
SET DBNAME=[削除（抹消）するデータベース名 ※既定ではirepodb]
SET LOGPATH=[削除（抹消）ログファイルのフルパス]
SET PORT=[PostgreSQLのポート番号 ※既定では5432]

"%DROPPGPATH%" -h localhost -U postgres -p %PORT% %DBNAME% > %LOGPATH% 2>&1
echo DropDB ErrorLevel=%ERRORLEVEL% >> %LOGPATH%
```

注意点

- ・データベースの削除（抹消）であるため実行すると、復元ができません。
バックアップが必ず取得できていることを確認してから実行してください。
- ・実行後、削除（抹消）ログファイルに"There is 1 other session using the database."の記載がある場合は、アプリケーションまたはDB管理ツールが接続状態であることが原因です。
IISアプリケーションの場合は、アプリケーションプールのリサイクル、DB管理ツールの場合は切断してから再度実行します。
- ・実行後、削除（抹消）ログファイルをメモ帳で開き"errorlevel=0"であることを確認します。

DB生成バッチサンプル

```
chcp 65001
SET PGPASSWORD=[PostgreSQLパスワード]
SET PGCREATEPATH=[createdb.exeまでのフルパス]
SET DBNAME=[生成するデータベース名 ※既定ではirepodb]
SET LOGPATH=[DB生成ログファイルのフルパス]
SET PORT=[PostgreSQLのポート番号 ※既定では5432]

"%PGCREATEPATH%" -h localhost -U postgres -p %port% %DBNAME% >> %LOGPATH% 2>&1
echo Create ErrorLevel=%ERRORLEVEL% >> %LOGPATH%
```

注意点)

- ・実行後、DB生成ログファイルをメモ帳で開き"errorlevel=0"であることを確認します。

DBリストアバッチサンプル

```
chcp 65001
SET PGPASSWORD=[PostgreSQLパスワード]
SET PGPATH=[pg_restore.exeまでのフルパス]
SET RESTOREDB=[リストア先のデータベース ※既定はirepodb]
SET DUMPPATH=[リストアするダンプファイルフルパス]
SET LOGPATH=[リストアログファイルのフルパス]
SET PORT=[PostgreSQLのポート番号 ※既定では5432]

"%PGPATH%" -h localhost -U postgres -p %port% -d %RESTOREDB% %DUMPPATH% > %LOGPATH% 2>&1
echo Restore ErrorLevel=%ERRORLEVEL% >> %LOGPATH%
```

注意点)

- ・リストア先のデータベースが空であることを確認して実行してください。
もしミスなどで再度リストアする場合は、DB削除（抹消）→生成を行った後に、実施します。
- ・実行後、リストアログファイルをメモ帳で開き"errorlevel=0"であることを確認します。