

AtScopeExeによる Jenkinsを使用したCI環境構築の手順

ビースラッシュ株式会社



1. AtScopeの2つの動作モード

- EAアドイン実行とコマンドライン実行の動作モードがあります
- 本書はコマンドライン実行の説明書です
 - コマンドライン実行では、リファクタリングスコアのcsvファイルを作成します
 - C言語のみ対応しています

2. コマンドライン実行AtScopeExe.exeの使い方

2-1. インストール

2-2. ライセンスキーと利用コードの入力

2-3. AtScopeExe.exeを実行

2-4. 出力ファイルの確認

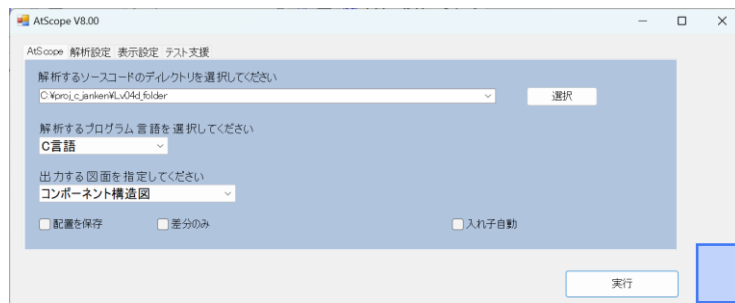
参考. Jenkinsとの連携例

1. AtScopeの2つの動作モード

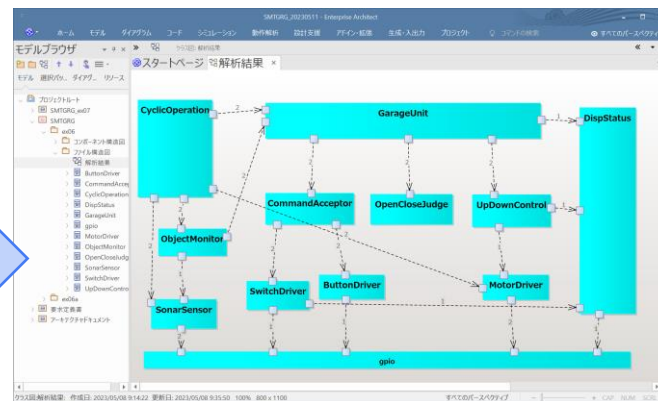


AtScopeV8の2つの動作モード

■ EAアドインとして動作するAtScope.dll



解析



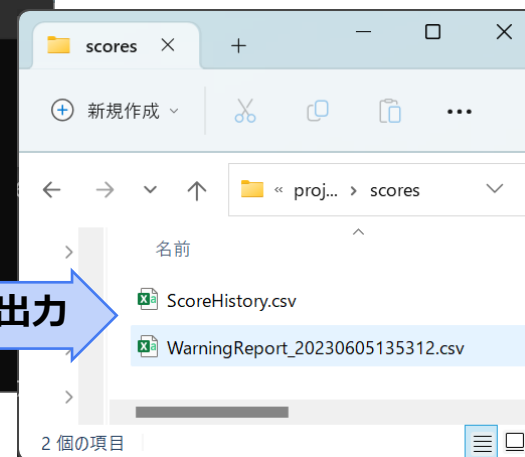
■ コマンドラインで動作するAtScopeExe.exe

```
コマンド プロンプト
C:\Program Files (x86)\AtScope>AtScopeExe.exe c:/proj24/ex06 c:/proj24/scores
AtScope V8.00
解析対象パス : [c:/proj24/ex06]
結果出力先パス : [c:/proj24/scores]
リファクタリングスコア・複雑度CSV出力
解析対象ファイル数 :38

Finished !

C:\Program Files (x86)\AtScope>
```

ファイル出力



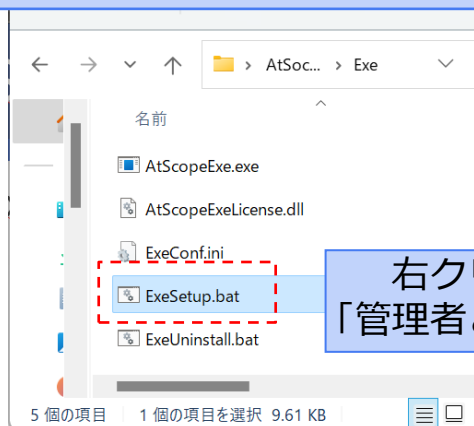
2. コマンドライン実行 AtScopeExe.exeの使い方



2-1. AtScopeExeのインストール

1. ExeSetup.bat を管理者権限で実行します

AtScope_Setup.zipを解凍し
ExeフォルダのExeSetup.bat
を管理者権限で実行



右クリックで
「管理者として実行」

```
C:\WINDOWS\System32\cmd.exe
本バッチの実行には、管理者権限が必要です。管理者権限かどうかを確認します。。。
管理者権限であることを確認しました。実行を続けます。

---『AtScopeExe』の付属フォルダAtScope をコピーします。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルをコピーしました。
1 個のファイルを移動しました。
1 個のファイルを移動しました。
1 個のファイルを移動しました。
1 個のファイルを移動しました。

---『AtScopeExe』の付属ファイルAtScopeExe.exe をコピーします。
1 個のファイルをコピーしました。
1 個のファイルを移動しました。

---『AtScopeExe』の付属ファイルAtScopeExeLicense.dll をコピーします。
1 個のファイルをコピーしました。
1 個のファイルを移動しました。

『AtScopeExe』のInstall が正常に完了しました。
続行するには何かキーを押してください . . .
```

2. 下記フォルダにインストールされます

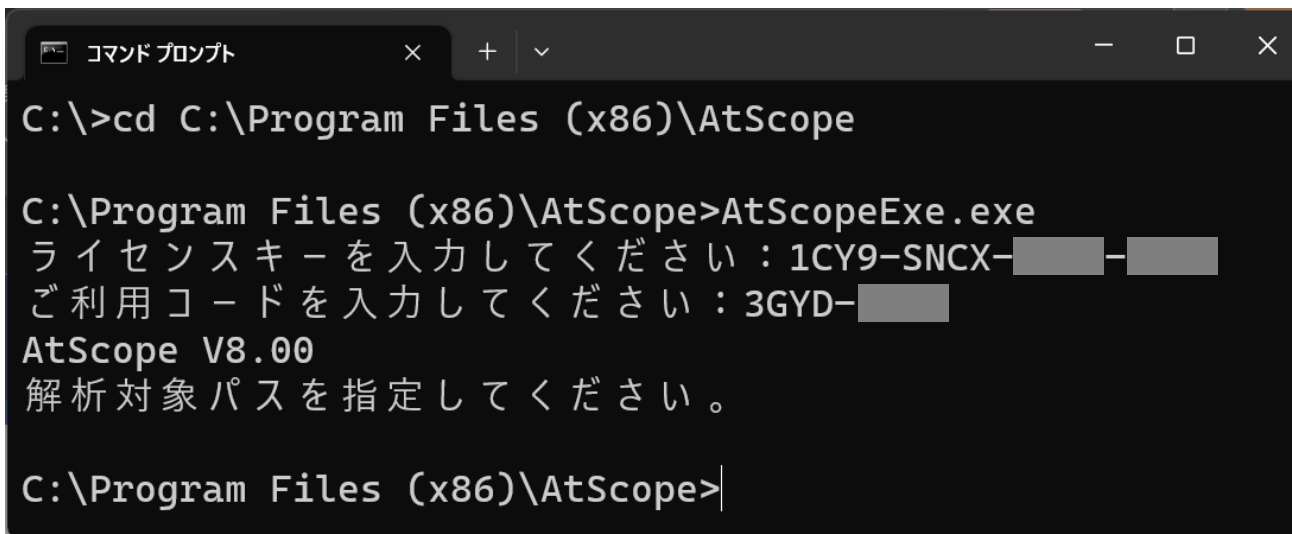
32bit版 Windows C:\Program Files\AtScope\AtScopeExe.exe

64bit版 Winodws C:\Program Files (x86)\AtScope\AtScopeExe.exe

※ アンインストールをするには ExeUninstall.bat を管理者権限で実行します

2-2. ライセンスキーと利用コードの入力

1. AtScopExe.exe を実行します
2. ライセンスキーを入力します
 - ハイフンありで入力してください
3. 利用コードを入力します



```
コマンド プロンプト
C:\>cd C:\Program Files (x86)\AtScope

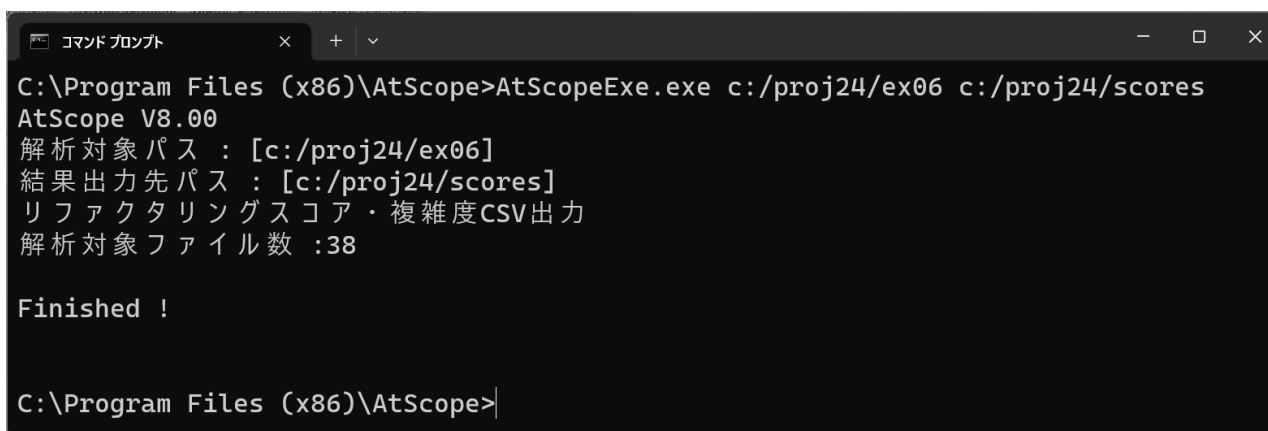
C:\Program Files (x86)\AtScope>AtScopeExe.exe
ライセンスキーを入力してください：1CY9-SNCX- - 
ご利用コードを入力してください：3GYD- 
AtScope V8.00
解析対象パスを指定してください。

C:\Program Files (x86)\AtScope>
```

2-3. AtScopeExeを実行

■ 実行コマンド

- AtScopeExeで解析設定パス（入力パスと出力パス）を指定してください
- **AtScopeExe.exe input_path output_path**



```
コマンド プロンプト
C:\Program Files (x86)\AtScope>AtScopeExe.exe c:/proj24/ex06 c:/proj24/scores
AtScope V8.00
解析対象パス : [c:/proj24/ex06]
結果出力先パス : [c:/proj24/scores]
リファクタリングスコア・複雑度CSV出力
解析対象ファイル数 : 38

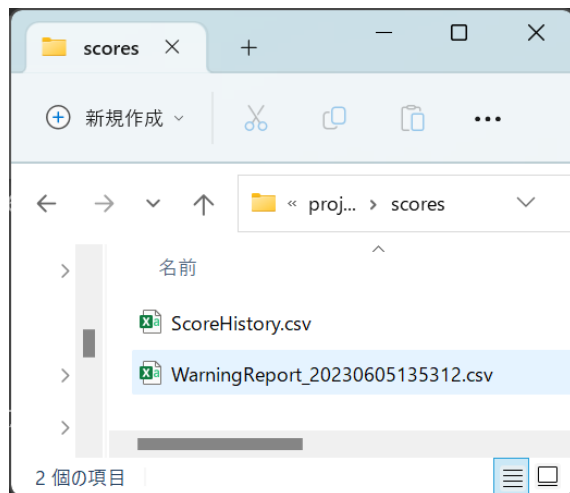
Finished !

C:\Program Files (x86)\AtScope>
```

- Jenkins等から起動する場合はサイレントオプション "-s" を指定してください
- **AtScopeExe.exe -s input_path output_path**

2-4.出力ファイルの確認

- 出力パスで指定したフォルダに2つのファイルができます

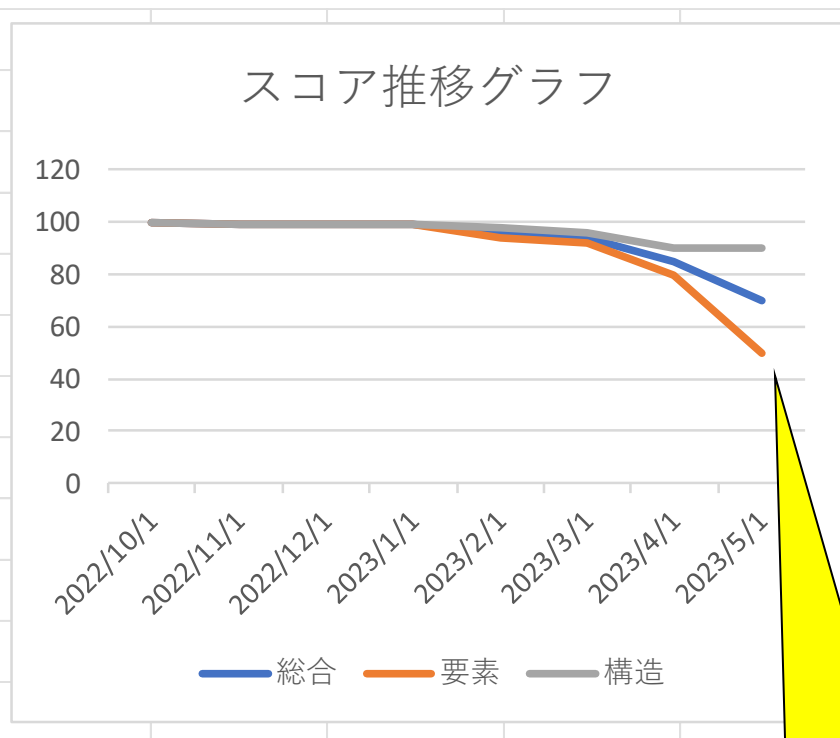


ファイル	内容
スコアファイル ScoreHistory.csv	実行毎に以下の項目を追記出力します 日時,総合点,要素点,構造点 ※AtScopeExeの実行毎に 1 行追加され、グラフ化することで時系列変化を見ることが出来ます
警告ファイル WarningReport_日時.csv ※日時はyyyyMMddmmHHss形式	減点対象のファイル名や関数名をします ・ ファイルの長さ／関数の数／関数の長さ ・ ファンアウト数／循環依存 ※AtScopeExeの実行毎に 1 ファイル生成されます

出力例と利用方法：ScoreHistory.csv

- ScoreHistory.csv をExcelに取り込み、グラフ化することでコード品質を可視化できます

日時	総合	要素	構造
2022/10/10 18:44	100	100	100
2022/11/10 18:46	99	99	99
2022/12/10 18:47	99	99	99
2023/1/10 18:48	99	99	99
2023/2/10 18:48	96	94	98
2023/3/10 18:49	94	92	96
2023/4/10 18:49	85	80	90
2023/5/10 18:50	70	50	90



要素点が低下
→関数やファイルが
肥大化しつつある

- WarningReport_日時.csv の例

WarningReport_20221110184634.csv

関数の長さ	GarageUnit gu_run()	59
ファンアウト数	CyclicOperation.c	13

WarningReport_20230510185032.csv

関数の長さ	ObjectMonitor om_monObject()	35
ファンアウト数	CyclicOperation.c	16
循環依存	2 : UpDownControl 2 : GarageUnit	

減点個所を把握して、
改善の方針が出せます

- AtScopeExeのエラーは以下の通りです

終了コード	意味
0	正常終了
-1	ライセンスエラー
-2	ライセンス有効期限切れ
-11	解析対象パス未指定
-12	結果出力先パス未指定
0x0A01	Parserエラー：ブレイスの数が不一致

参考. Jenkinsとの連携例



- コマンドラインツール AtScopeExeをJenkins のプロジェクトとして実行することができます
- ソースコードの劣化を防止しながら継続的インテグレーションを行う環境を構築します

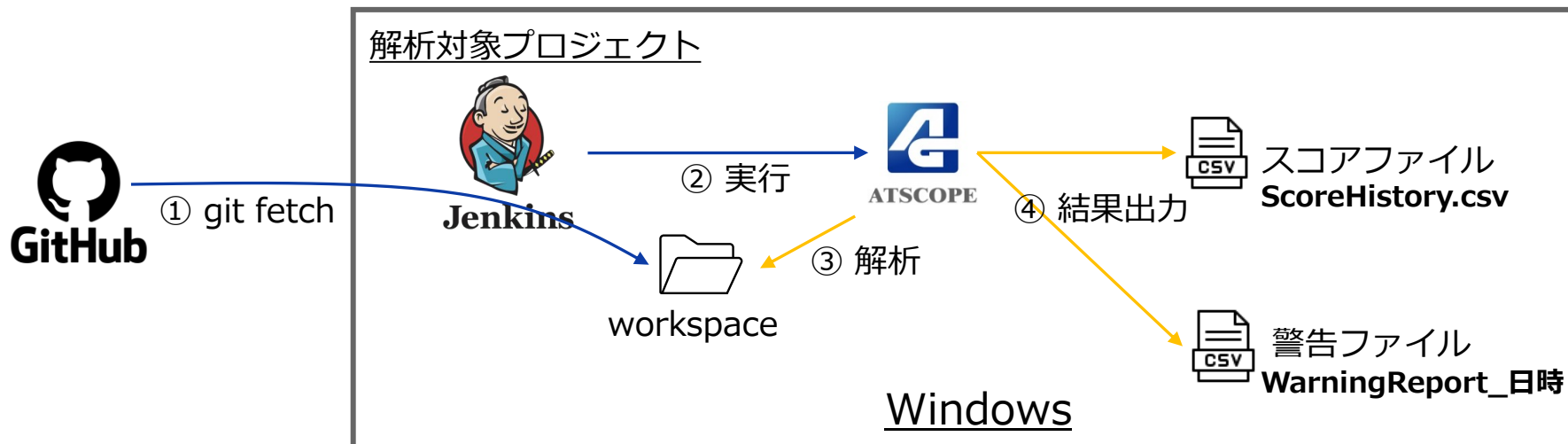


AtScopeExeは、ソースコードのスコア（総合点、要素点、構造展）および、ソースコードのレポート（ファイルの長さ、関数の数、関数の長さ、ファンアウト数、循環依存）を算出し、CSVファイルとして出力するWindowsのコマンドラインツールです。



Jenkins(ジェンキンス) は、継続的インテグレーション(CI)や継続的デリバリー(CD)、継続的デプロイメントを実現するOSSのツールです。

- CIツールとしてJenkins、SCMとしてGitHubを使用した例です
 - AtScopeExeはJenkinsが稼働するWindows上で実行します



- AtScopeExeは2つのCVSファイルを出力します

種別	ファイル名	内容
スコアファイル	ScoreHistory.csv	構造スコア 100点満点での減点方式
警告ファイル	WarningReport_日時.csv 日時はyyyyMMddmmHHss形式日時	減点されている箇所

AtScopeExeの起動バッチを作成する



- AtScopeExeの起動バッチを作成します。起動バッチはJenkinsから呼び出します
- Jenkinsのプロジェクト 1 つにつき、1 つの起動バッチを用意します
- 解析対象のフォルダはJenkinsのプロジェクトのワークスペースフォルダ以下になりますので、Jenkinsのプロジェクト設定（プロジェクト名）に依存します
- 結果出力先のフォルダは任意のフォルダとします。
- 起動バッチはJenkins実行ユーザーがアクセス可能な任意のフォルダに配置します

項目	内容
バッチファイル名	AtScope_curl.bat
バッチファイルパス	C:¥Program Files (x86)¥AtScope¥bat¥AtScope_curl.bat
解析対象パス	C:¥ProgramData¥Jenkins¥.jenkins¥workspace¥curl¥src
結果出力先パス	C:¥share¥AtScope_Score¥curl

Jenkinsにジョブを作成する（１）

- 以下のジョブ（＝プロジェクト）をJenkinsに作成します。今回はソースコードのビルドは行わず、AtScopeExeの実行のみを行うジョブとします

項目	内容
ジョブ名	curl
SCM	GitHub
リポジトリURL	https://github.com/curl/curl.git
ビルドトリガ	1 時間ごとのポーリング ポーリング時に変更を検出した場合にビルドが実行されます
バッチコマンド	C:¥Program Files (x86)¥AtScope¥bat¥AtScope_curl.bat

Jenkinsにジョブを作成する（２）

- Jenkinsにログインし「ダッシュボード」から「+新規ジョブ作成」をクリックします



The screenshot shows the Jenkins dashboard interface. At the top is a navigation bar with the Jenkins logo, a search bar, and user information for Kazuma Sonoda. Below the navigation bar is a breadcrumb trail for the dashboard. On the left sidebar, the '+ 新規ジョブ作成' (New Job) button is highlighted with a red rectangle. The main content area displays a table of existing jobs. The table has columns for status (S), workspace (W), name, latest successful build, latest failed build, and build time. Two jobs are listed: 'AtScopeTest_1' and 'tmp'.

S	W	名前 ↓	最新の成功ビルド	最新の失敗ビルド	ビルド所要時間
✓	☁	AtScopeTest_1	7 日 1 時間 #34	7 日 7 時間 #32	4.2 秒
✓	☁	tmp	22 分 #24	12 時間 #22	4 秒

Jenkinsにジョブを作成する（3）

- ジョブ名（今回は curl とする）を入力し、「フリースタイル・プロジェクトのビルド」を選択し「OK」をクリックします



The screenshot shows the Jenkins web interface. At the top is a navigation bar with the Jenkins logo, a search bar, and user information for Kazuma Sonoda. Below the navigation bar is a breadcrumb trail: ダッシュボード > すべて >. The main content area is titled 'ジョブ名入力' (Job Name Input). It features a text input field containing 'curl', which is highlighted with a red rectangle. Below the input field is a label '必須項目' (Required field). Below the input field are three selectable options, each with an icon and a description. The first option, 'フリースタイル・プロジェクトのビルド' (Build a Freestyle Project), is highlighted with a red rectangle. The second option is 'パイプライン' (Pipeline). The third option is 'マルチ構成プロジェクトのビルド' (Build a Multibranch Project). At the bottom of the form is a blue 'OK' button, which is also highlighted with a red rectangle.

ジョブ名入力

curl

» 必須項目

 **フリースタイル・プロジェクトのビルド**
もっとも汎用性の高いJenkinsの中核機能です。任意のSCMからソースコードをチェックアウトし、任意のビルドシステムでプロジェクトがビルドできます。往々にして、ソフトウェアのビルド以外にも様々な仕事の自動化に利用することができます。

 **パイプライン**
複数のビルドスレーブにまたがる長時間に渡る処理を編成します。（以前は、ワークフローとして知られていた）パイプラインの構築に適しており、フリースタイルジョブでは上手く扱えない複雑な処理を容易に編成することができます。

 **マルチ構成プロジェクトのビルド**
テストや、プラットフォームごとのビルドなどといった、多数の異なる構成が必要なプロジェクトに適しています。

OK

Multibranch Pipeline

Jenkinsにジョブを作成する（４）

- リポジトリURL（今回は `https://github.com/curl/curl.git` とする）を入力します

ダッシュボード > curl >

General ソースコード管理 ビルド・トリガ ビルド環境 ビルド ビルド後の処理

☐ なし

☒ Git ?

リポジトリ ?

リポジトリURL ? ×

`https://github.com/curl/curl.git`

認証情報 ?

- なし

+ 追加

高度な設定...

保存 Apply

Jenkinsにジョブを作成する（5）

- 「ビルド・トリガ」でビルド（AtScopeExeを実行）するタイミングを定義します。今回は1時間ごとにSCM（GitHub）をポーリングし、更新があった場合にビルドを実行します

ダッシュボード > curl >

General ソースコード管理 **ビルド・トリガ** ビルド環境 ビルド ビルド後の処理

☐ リモートからビルド (例: スクリプトから) ?

☐ GitHub hook trigger for GITScm polling ?

☒ **SCMをポーリング** ?

スケジュール ?

H * * * *

前回は2022年10月19日水曜日 17時23分02秒 日本標準時に起動しました。次は2022年10月19日水曜日 18時23分02秒 日本標準時に起動します。

☐ post-commitフックを無視 ?

☐ 他プロジェクトの後にビルド ?

☐ 定期的に行う ?

保存 Apply

Jenkinsにジョブを作成する（6）

- 「ビルド手順の追加」 から「Windowsバッチコマンドの実行」を選択します



Jenkinsにジョブを作成する（7）

- 「ビルド手順の追加」から「Windowsバッチコマンドの実行」を選択し、作成した起動バッチのパスを設定後「保存」します



ダッシュボード > curl >

General ソースコード管理 ビルド・トリガ ビルド環境 **ビルド** ビルド後の処理

☐ With Ant ?

☐ コンソール出力にタイムスタンプを追加する

☐ 滞留した場合にビルドを中止する

ビルド

ビルド手順の追加 ▲

Filter

- Antの呼び出し
- Invoke Gradle script
- Mavenの呼び出し
- Set build status to "pending" on GitHub commit
- Windows/バッチコマンドの実行**
- シェルの実行
- タイムアウト付きの実行

Windows/バッチコマンドの実行 ?

コマンド

ビルドから利用可能な環境変数の一覧

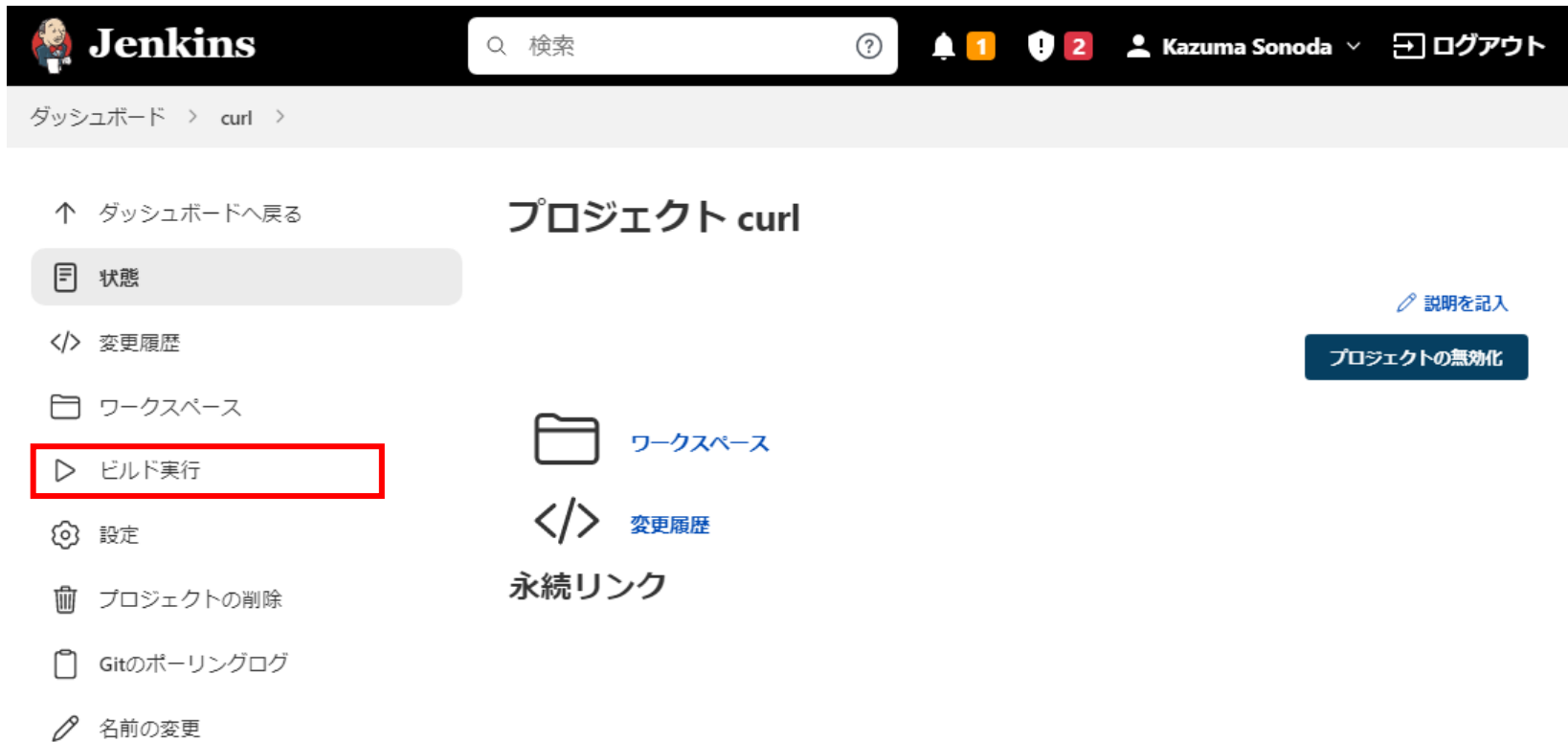
"C:\Program Files (x86)\AtScope\bat\AtScope_curl.bat"

高度な設定...

保存 Apply

Jenkinsにジョブを作成する（8）

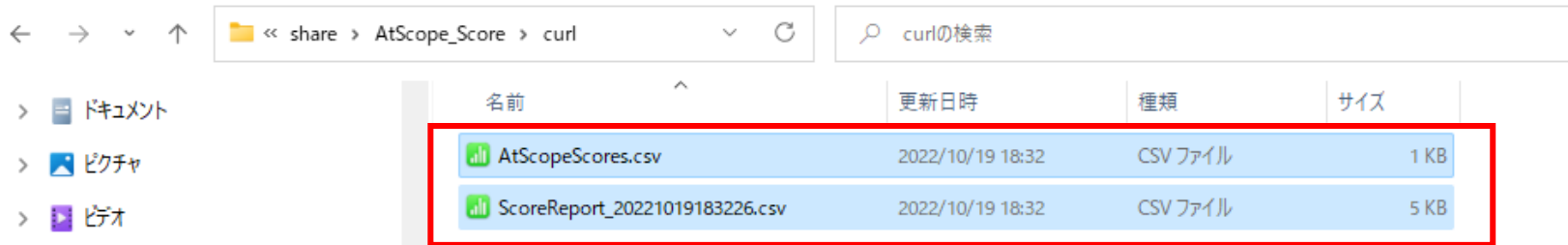
- プロジェクト「curl」が作成されました。この時点ではワークスペース（gitリポジトリを fetch するフォルダ）は作成されていないので、一度、手動で「ビルド実行」をします



The screenshot shows the Jenkins web interface for a project named 'curl'. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Kazuma Sonoda'. The breadcrumb trail indicates the current location is 'ダッシュボード > curl >'. On the left sidebar, the 'ビルド実行' (Build) option is highlighted with a red rectangle. The main content area displays the project name 'プロジェクト curl' and a list of actions: 'ワークスペース' (Workspace), '変更履歴' (Change History), and '永続リンク' (Permanent Link). A button labeled '説明を記入' (Add Description) is visible on the right. The bottom of the page shows the page number '24' and the copyright notice 'Copyright 2023 BACKSLASH DESIGN Co., Ltd.'.

Jenkinsにジョブを作成する（９）

- ビルドが実行されると、ワークスペースがつくられgitリポジトリが fetch された後、起動バッチが実行されます
- ビルドが成功すると、起動バッチで指定した出力フォルダ（今回は C:¥share¥AtScope_Score¥curl ）にCSVファイルが作られます



名前	更新日時	種類	サイズ
AtScopeScores.csv	2022/10/19 18:32	CSV ファイル	1 KB
ScoreReport_20221019183226.csv	2022/10/19 18:32	CSV ファイル	5 KB

- 以降は、「ビルド・トリガ」で指定したタイミングでビルドが実行されます

Jenkinsにジョブを作成する（10）

- ジョブが正常終了しなかった場合は、コンソール出力を確認します。
AtScopeExeがエラーの場合はエラーコード（p.12）を確認してください



Jenkins

ダッシュボード > curl > #12

↑ プロジェクトへ戻る

目 状態

</> 変更履歴

📄 コンソール出力

📄 プレーンテキスト表示

⚙ 説明の編集

🗑 Delete build '#12'

🔍 Git Build Data

← 前のビルド

✖ コンソール出力

ユーザー-Kazuma Sonodaが実行

Running as SYSTEM

ビルドします。 ワークスペース: C:\ProgramData\Jenkins\workspace\curl

The recommended git tool is: NONE

No credentials specified

```
> C:\Program Files\Git\cmd\git.exe rev-parse --resolve-git-dir
C:\ProgramData\Jenkins\workspace\curl\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\cmd\git.exe config remote.origin.url https://github.com/curl/curl.git # timeout=10
Fetching upstream changes from https://github.com/curl/curl.git
> C:\Program Files\Git\cmd\git.exe --version # timeout=10
> git --version # 'git version 2.37.1.windows.1'
> C:\Program Files\Git\cmd\git.exe fetch --tags --force --progress -- https://github.com/curl/curl.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\cmd\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 81094cb49256dec3bdb94052f8542b15d54e4a1 (refs/remotes/origin/master)
> C:\Program Files\Git\cmd\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\cmd\git.exe checkout -f 81094cb49256dec3bdb94052f8542b15d54e4a1 # timeout=10
Commit message: "tests: avoid CreateThread if _beginthreadex is available"
> C:\Program Files\Git\cmd\git.exe rev-list --no-walk 81094cb49256dec3bdb94052f8542b15d54e4a1 #
timeout=10
[curl] $ cmd /c call C:\Users\sonoda\AppData\Local\Temp\jenkins15948767925734576509.bat

C:\ProgramData\Jenkins\workspace\curl>"C:\Program Files (x86)\AtScope\bat\AtScope_curl.bat"
終了コード: -3
build step 'windows/バッチコマンドの実行' marked build as failure
Finished: FAILURE
```

これで本チュートリアルは終了です

AtScopeExeをJenkinsに組み込み、是非、ソースコードの品質の“維持”“向上”にお役立てください！