



GET STARTED WITH METAVERSION

ユーザガイド V2

もくじ

- ① はじめに
- ② セットアップ
- ③ 管理者向け機能
- ④ ユーザ向け機能



はじめに |

METAVERSIONとは？

MetaVersionはSVNリポジトリによりバージョン管理を行っているシステム開発を支援するツールです。

このツールは各プロジェクトによるコミットの情報記録し、同一期間に同一パスに対する変更を行ったプロジェクトを把握することを容易にします。

また履歴表示や差分表示に相応のレスポンスタイムを必要とするSVNコマンドに対する部分的な代替手段を提供します。

どうしてこのようなツールが必要なのか？

システム開発の現場ではしばしば複数のプロジェクトが並行して進んでいます。

保守運用フェーズともなれば比較的大規模な開発プロジェクトだけでなく、小規模で期間も短いシステム改修も多く発生します。

こうした並行開発状況においては、各プロジェクトにおいてどのようなコミットを行ってきたか把握することが必要になります。

またとくに複数プロジェクト間で同一パスに対する変更が発生していないかを確認することが必要になります。このようなニーズに応えるため MetaVersion は構想されました。

機能概要

コミット自動取込み

- 登録されたリポジトリ情報に基づき、コミット情報を定期的に自動取込み。

コミット-プロジェクト紐付け

- コミット・メッセージに基づき、コミット情報とプロジェクト情報を自動紐付け
- 画面オペにより、コミット情報とプロジェクト情報を手動紐付け
- 画面オペにより、プロジェクトがコミットする予定のパスを手動紐付け

各種一覧表示とCSVダウンロード

- プロジェクトが変更したパスの一覧
- 同一パスを変更している他プロジェクトの一覧
- プロジェクトがコミットする予定のパスの一覧



セッ アッ プ

ミドルウェアの導入

MetaVersionのセットアップには次のミドルウェアが必要です：

- JDK バージョン7以上
- Maven バージョン3以上
- PostgreSQL バージョン 9.4以上
- Tomcat バージョン7以上

セットアップ 動作確認まで

- ① GitHubからプロジェクトのファイル一式をダウンロード
- ② "ext/ddl"配下のSQLファイルを実行してDBオブジェクトを作成
- ③ "src/resources/application.properties"のDB接続情報を適宜変更シェルで"mvn clean spring-boot:run"を実行
- ④ シェルに"Started MVApplication in ... seconds"という文言が出力されるのを待機
- ⑤ Webブラウザで"http://localhost:8080/init"にアクセスしマスタデータと初期ユーザadministratorの作成を実行
- ⑥ 初期化が成功した旨のメッセージが表示されるのを確認

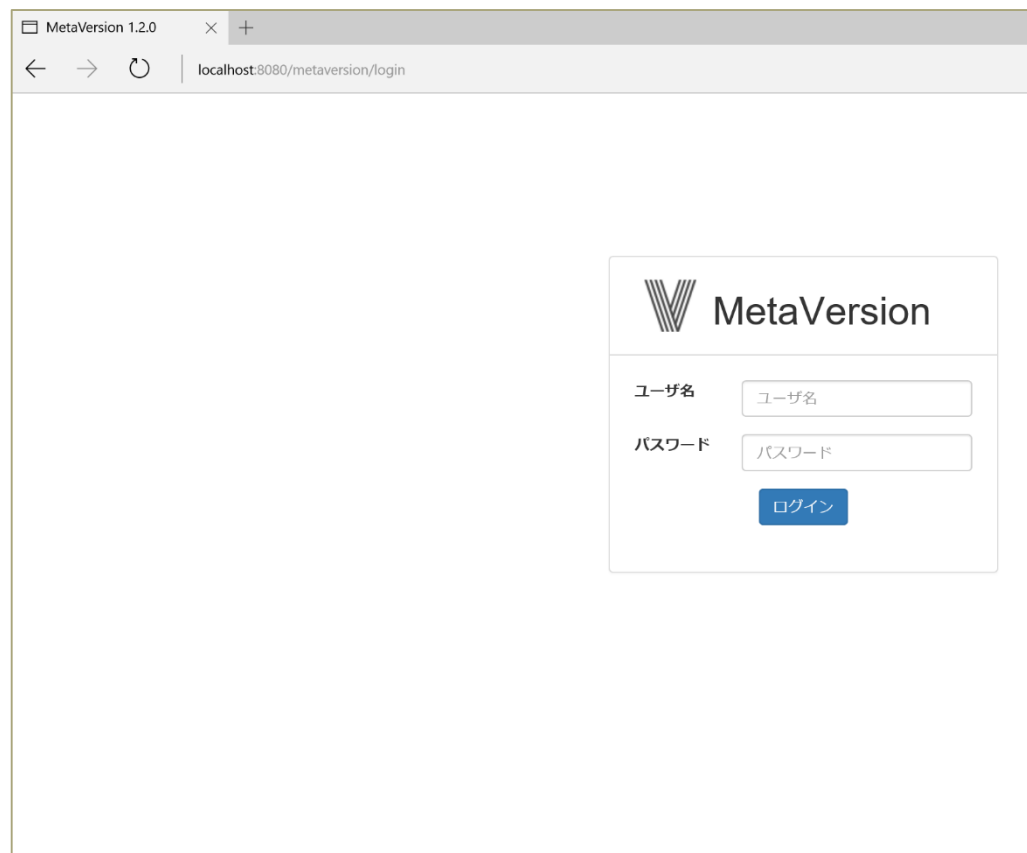
セットアップ 正式のデプロイ まで

- ① プロジェクトのルート・ディレクトリで"mvn clean package"を実行
- ② 作成されたmetaversion.warファイルをTomcatのwebappsディレクトリにコピー
- ③ Webブラウザで"http://localhost:8080/metaversion"にアクセスし、ログイン画面が表示されるのを確認
- ④ 同画面で"application.properties"に記載された初期ユーザのログイン情報を入力しログイン

ログイン画面

認証を済ませていない状態で
MetaVersionの各種画面のURLにア
クセスすると、ログイン画面にリダ
イレクトされます。

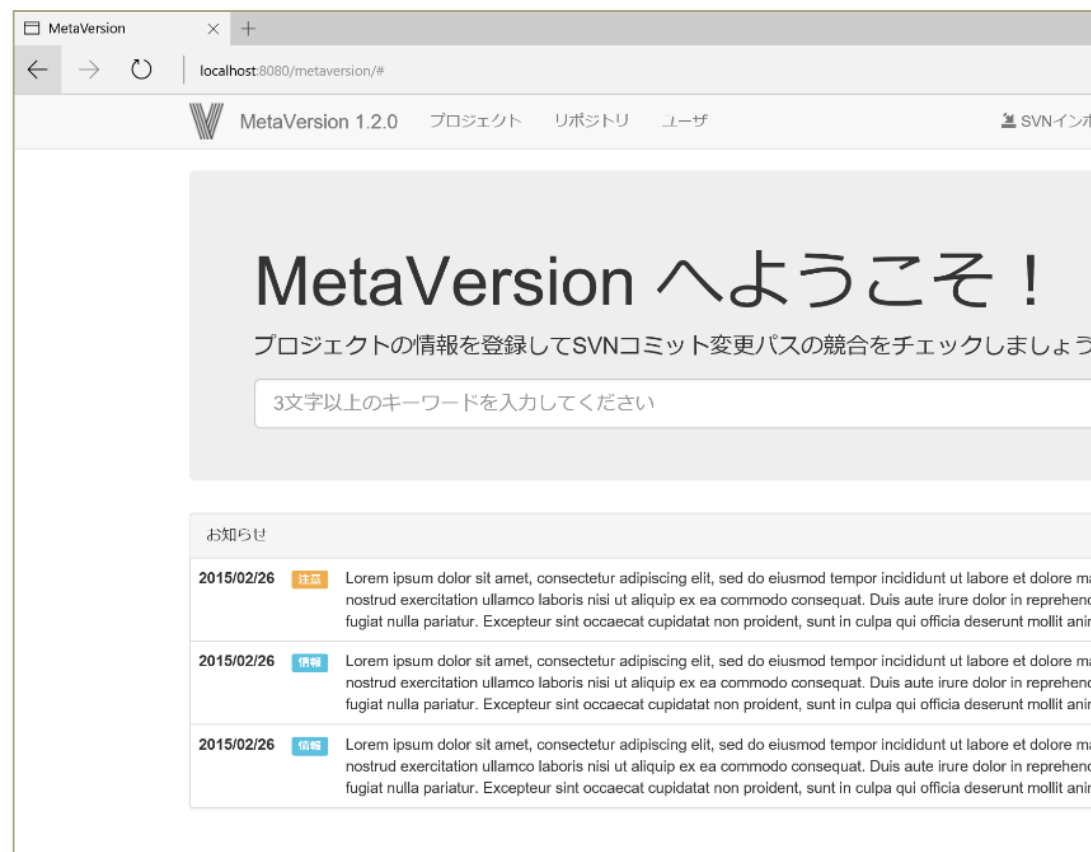
セットアップ直後は初期ユーザ
administratorでしかログインでき
ません。後述の管理者機能でユーザ
を追加してください。



トップ画面

ログインを済ませるとトップ画面に遷移します。

このページからプロジェクト一覧、リポジトリ一覧、ユーザー一覧の3画面に移動できます。



カスタマイズの方法

□DB接続先を変更したい

- "src/resources/application.properties"に記載されたJDBC接続文字列や接続ユーザ情報を適宜変更してください。

□ログ出力設定を変更したい

- "src/resources/log4j.properties"に記載されたLoggerやAppenderの設定を適宜変更してください。

□初期ユーザのパスワードを変更したい

- 後述の管理者向け機能を使って変更してください。

□コンテキストパス名を変更したい

- mvn packageコマンドで生成されたwarファイル名を変更するか、pom.xmlファイル内のパッケージ・ファイル名の指定箇所を変更してから再ビルドしてください。

□トップ画面のお知らせ欄を更新したい

- 残念ながらバージョン1.2時点ではこの欄に表示された情報を更新するための画面はありません。
- "src/main/resources/static/js/templates/index.html"を編集し、リビルド、デプロイを行ってください。

アップグレード方法 1.1→1.2

DBオブジェクトのうち次のものを新規作成/再作成してください：

- "virtual_changed_path"テーブルおよびその制約 新規作成
- "project_changedpath_plus_view"ビュー 新規作成
- "virtual_changed_path_seq"シーケンス 新規作成
- "project_parallels_view"ビュー 再作成

※"ext/ddl"配下に移行用スクリプトを用意してあります。

アップグレード方法 1.2→1.3

DBオブジェクトのうち次のものを新規作成/再作成してください：

- "project"テーブル カラム追加
- "project_path_count_view"ビュー 新規作成
- "project_stats_view"ビュー 再作成
- "project_changedpath_view"ビュー 再作成
- "project_changedpath_plus_view"ビュー 再作成
- "project_parallel_view"ビュー 再作成

※"ext/ddl"配下に移行用スクリプトを用意してあります。



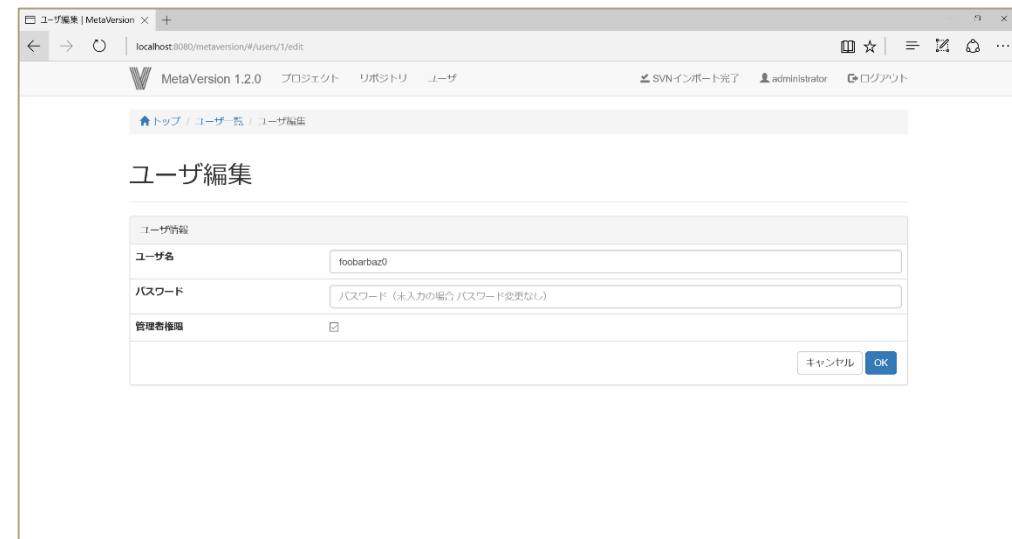
管理者向け機能


ユーザ管理

メニューの「ユーザ」をクリックするとユーザー一覧が表示されます。

一覧右上・右下の「追加」をクリックするとユーザ追加画面が表示されます。

一覧されたユーザの「編集」をクリックするとユーザ情報の更新が行えます。





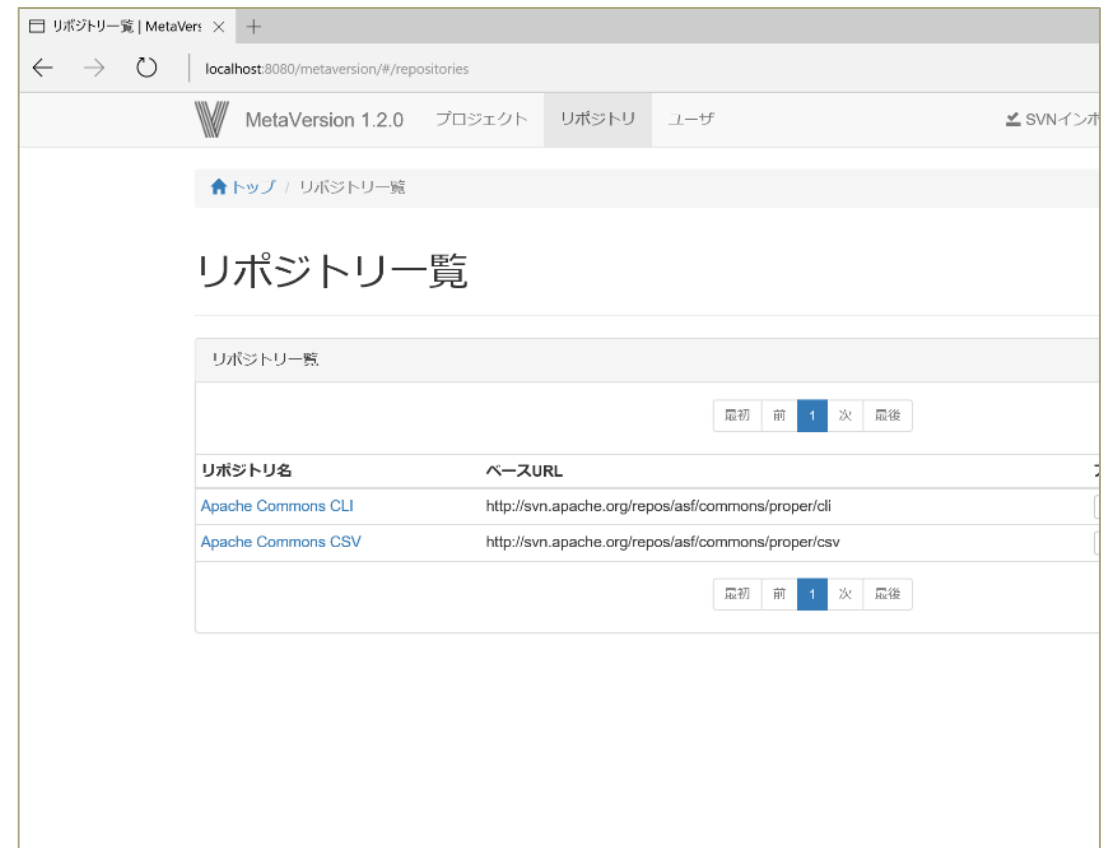
ユ一ザ向け機能

リポジトリ一覧

メニューの「リポジトリ」をクリックするとリポジトリ一覧が表示されます。

一覧には登録済みのSVNリポジトリ情報が表示されます。

ここからリポジトリ追加画面や詳細画面に移動できます。

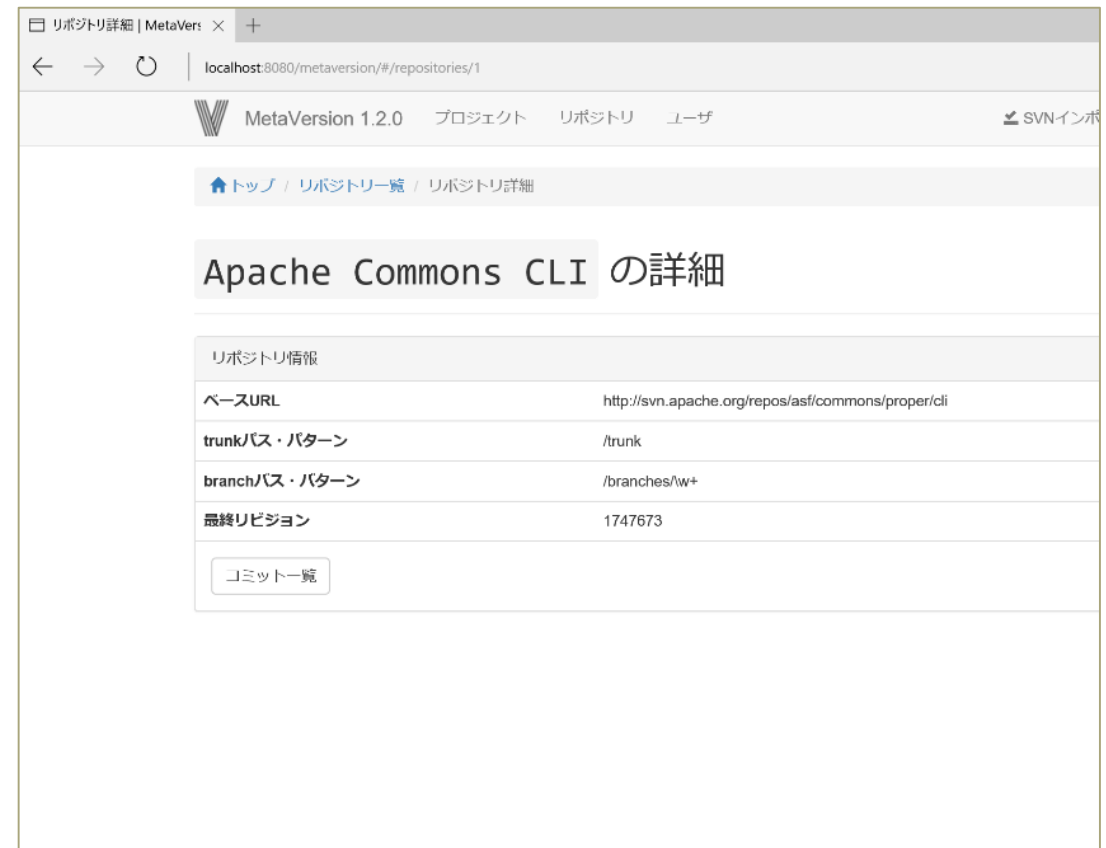


リポジトリ詳細画面

SVNリポジトリの情報を参照できます。

〔編集〕ボタンをクリックすることでリポジトリ編集画面に移動できます。

〔最終リビジョン〕はMetaVersionへの取込みが終わっているコミットの最終リビジョンを指します。



The screenshot displays the 'リポジトリ詳細' (Repository Details) page in the MetaVersion 1.2.0 application. The browser address bar shows 'localhost:8080/metaversion/#/repositories/1'. The page header includes navigation links for 'プロジェクト' (Project), 'リポジトリ' (Repository), and 'ユーザ' (User), along with an 'SVNインポ' (SVN Import) button. The breadcrumb trail is 'トップ / リポジトリ一覧 / リポジトリ詳細'. The main title is 'Apache Commons CLI の詳細'. Below this, a table provides repository information:

リポジトリ情報	
ベースURL	http://svn.apache.org/repos/asf/commons/proper/cli
trunkパス・パターン	/trunk
branchパス・パターン	/branches/w+
最終リビジョン	1747673

At the bottom of the table is a button labeled 'コミット一覧' (Commit List).

リポジトリ追加/編集画面

「ベースURL」はリポジトリのルートである必要はありません。後述のパス・パターンと組み合わせて取り込み対象としたいURLとなるよう指定します（※）。

「...パス・パターン」には「ベースURL」配下で「この接頭辞に一致したらtrunk/branchとみなす」という正規表現を指定します。

接続先リポジトリが一般に公開されている場合は「接続ユーザ名」とパスワードは空欄のままでOKです。

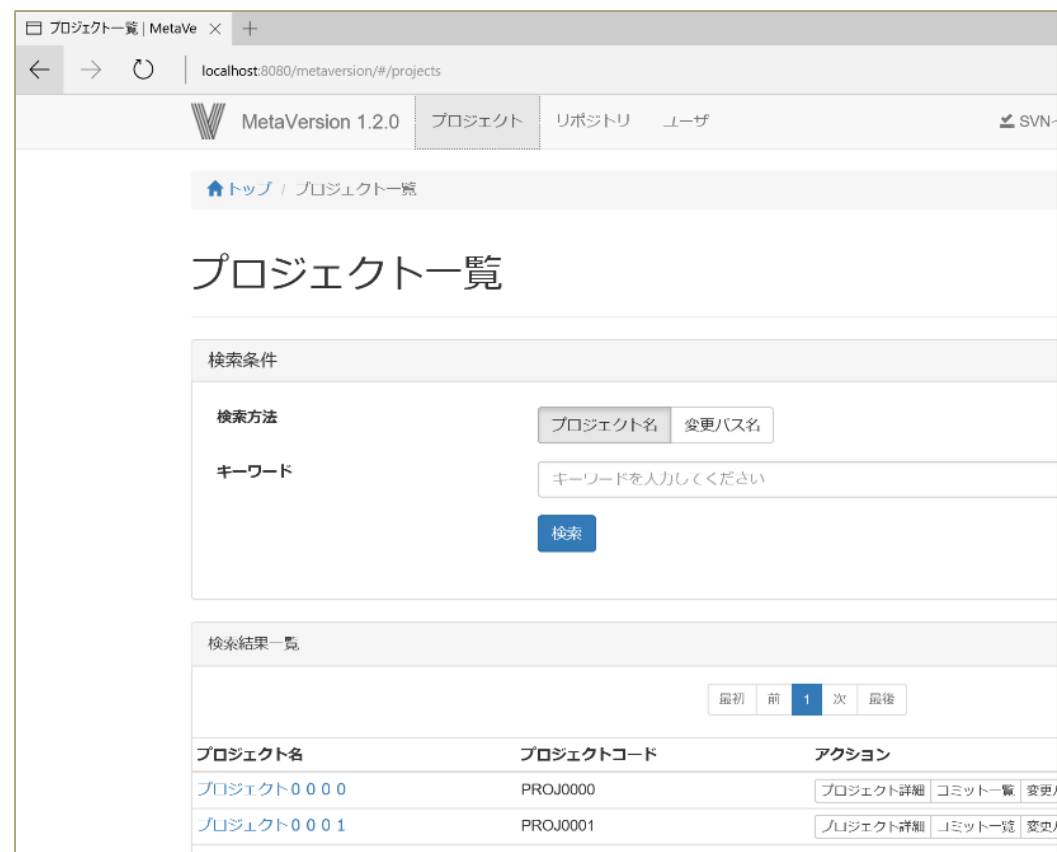
※右図にも示したApache CommonsのSVNはまさに好例です。このリポジトリはルートの下に複数のサブディレクトリがあり、その配下にCLIやCSVなどの個別のプロジェクトのツリーがあるので「ベースURL」には「http://svn.apache.org/repos/asf/commons/proper/cli」、[trunkパス・パターン]には「/trunk」を指定します。

プロジェクト一覧

登録済みプロジェクトの一覧が表示されます。

プロジェクト名（およびプロジェクト・コード）に基づく検索と、プロジェクトが変更したパス名に基づく検索ができます。

プロジェクト追加画面や詳細画面などへの移動もここから行うことができます。



プロジェクト詳細

「コミット・サイン」の正規表現パターンにマッチした文言がコミット・メッセージに含まれるSVNコミットは当該プロジェクトによるものとみなされ、自動で紐付けられます。

この画面から「コミット一覧」「変更パス一覧」「並行開発一覧」などの各種画面に移動できます。

「リリース予定日」は並行開発の判断ロジックで利用されます。

The screenshot shows the MetaVersion 1.2.0 web interface. The browser address bar shows 'localhost:8080/metaversion/#/projects/1'. The page title is 'プロジェクト0000の詳細' (Details of Project 0000). Below the title is a table of project information.

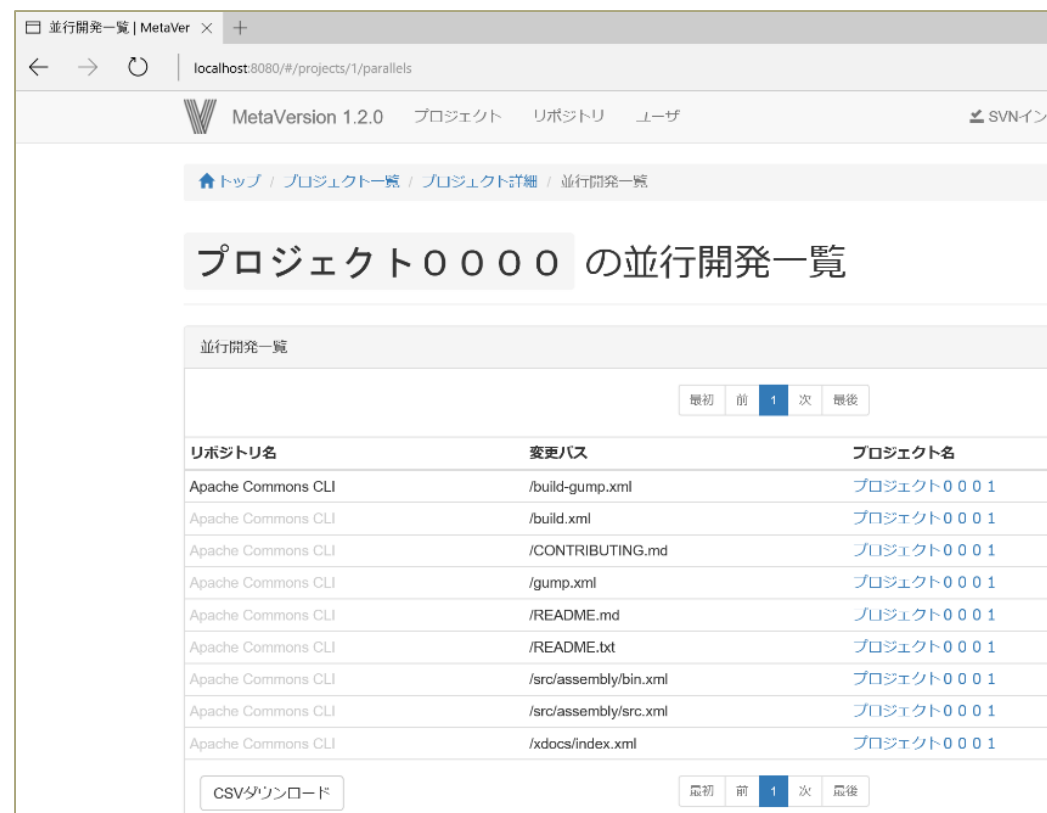
プロジェクト情報	
コード	PROJ0000
担当者	foo
コミット・サイン	\$proj0000\b
初期コミット日時	2007/07/29 15:42:06.543
最終コミット日時	2016/01/19 06:19:49.710
コミット回数	6
変更パス数	28

At the bottom of the table, there are four buttons: 'コミット一覧' (Commit List), '変更パス一覧' (Change Path List), '仮想変更パス一覧' (Virtual Change Path List), and '並行開発一覧' (Parallel Development List).

並行開発一覧

変更パスとそのパスを同時期に（※）コミットしている他プロジェクトが一覧されます。

パスには互いのプロジェクトの「コミット予定のパス」（仮想変更パス）も含まれます。



The screenshot shows the MetaVersion 1.2.0 web interface. The browser address bar shows 'localhost:8080/#/projects/1/parallels'. The page title is 'プロジェクト0000 の並行開発一覧'. Below the title, there is a table with the following columns: 'リポジトリ名' (Repository Name), '変更パス' (Change Path), and 'プロジェクト名' (Project Name). The table lists 10 entries, all with 'Apache Commons CLI' as the repository name and 'プロジェクト0000' as the project name. The change paths are: '/build-gump.xml', '/build.xml', '/CONTRIBUTING.md', '/gump.xml', '/README.md', '/README.txt', '/src/assembly/bin.xml', '/src/assembly/src.xml', and '/xdocs/index.xml'. At the bottom of the table, there is a 'CSVダウンロード' (Download CSV) button. Navigation buttons for '最初' (First), '前' (Previous), '1' (Current), '次' (Next), and '最後' (Last) are visible at the bottom of the table.

リポジトリ名	変更パス	プロジェクト名
Apache Commons CLI	/build-gump.xml	プロジェクト0000
Apache Commons CLI	/build.xml	プロジェクト0000
Apache Commons CLI	/CONTRIBUTING.md	プロジェクト0000
Apache Commons CLI	/gump.xml	プロジェクト0000
Apache Commons CLI	/README.md	プロジェクト0000
Apache Commons CLI	/README.txt	プロジェクト0000
Apache Commons CLI	/src/assembly/bin.xml	プロジェクト0000
Apache Commons CLI	/src/assembly/src.xml	プロジェクト0000
Apache Commons CLI	/xdocs/index.xml	プロジェクト0000

※「同時期」と判断するロジックについては次スライドを参照のこと。

「同時期」 判断ロジック

- 基本ロジック

- 「同時期」と判断するロジックは次の通り：「当該パスに関して、プロジェクトAによる最古のコミット日時と最新のコミット日時、プロジェクトBの最古のコミット日時と最新のコミット日時を求め、両期間の間に重なる部分があれば『同時期』とみなす」。

- 特殊ロジック① v1.2以降

- 「コミット予定のパス」にはコミット日時が存在しないので、かわりに「プロジェクトが紐付けられている最古のコミット（そこに含まれる変更パス名は問わない）と最新のコミット（"）」を求めて、そのパスの期間として利用します。

- 特殊ロジック② v1.3以降

- v1.3において「最古のコミット日時」の算定に、「リリース予定日」についての考慮が加わりました。
- 「プロジェクトに『リリース予定日』が設定されており かつ パスの最新コミット日時より『...予定日』が最新（未来）である場合、『最古のコミット日時』には『リリース予定日』が用いられる」。