

## Natural for Windows

### Natural スタジオの拡張

バージョン 6.3.3

October 2008

This document applies to Natural バージョン 6.3.3 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © Software AG 1992-2008. All rights reserved.

The name Software AG™, webMethods™, Adabas™, Natural™, ApplinX™, EntireX™ and/or all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

# 目次




1 Natural スタジオの拡張 .....	1
2 Object Compare .....	3
開始する前に .....	4
Object Compare の要素 .....	5
オプションの設定 .....	6
比較するオブジェクトの選択 .....	7
Object Compare の開始 .....	7
3 ローカルバージョン管理 .....	9
予備注釈 .....	10
開始する前に .....	11
ローカルバージョン管理の要素 .....	13
オプションの設定 .....	16
プロジェクトの管理 .....	20
ローカルバージョン管理のファーストステップ .....	27
ローカルバージョン管理の操作 .....	44
ログの処理 .....	55
コマンドスクリプト .....	56
4 Natural for Ajax Tools .....	57
開始する前に .....	58
Natural for Ajax Tools の要素 .....	59
サーバーの設定 .....	60
ライブラリとプロジェクトの設定 .....	61
Application Designer 開発ワークスペースを開く .....	61
生成されたアダプタのインポート .....	62
アダプタからのメインプログラムの生成 .....	63
ブラウザでのメインプログラムの実行 .....	63
Natural プログラムまたは Application Designer 開発ワークスペースの URL の 表示 .....	64
索引 .....	67



# 1 Natural スタジオの拡張

---

このドキュメントでは、Natural 付属の Natural スタジオの拡張機能について説明します。次の項目で構成されています。

 <b>Object Compare</b>	2つの異なる Natural ソースの内容を比較する方法について説明します。
 ローカルバージョン管理	ソース制御システムを操作する方法について説明します。
 <b>Natural for Ajax Tools</b>	Natural for Ajax 機能を Natural スタジオから直接使用する方法について説明します。



## 2 Object Compare

---

▪ 開始する前に .....	4
▪ Object Compare の要素 .....	5
▪ オプションの設定 .....	6
▪ 比較するオブジェクトの選択 .....	7
▪ Object Compare の開始 .....	7

Object Compare を使用すると、2つの異なる Natural ソースの内容を比較できます。

サポートされているオブジェクトタイプは、プログラム、ダイアログ、クラス、サブプログラム、サブルーチン、テキスト、ローカルデータエリア、パラメータデータエリア、グローバルデータエリア、ヘルプルーチン、コピーコード、機能、およびアダプタです。



**Note:** Object Compare プラグインの有効化/無効化に対する特別な Natural Security サポートはありません。ただし、Object Compare では Natural ソースの読み取りにシステムコマンド LIST を使用します。LIST コマンドに提供されるのと同じ Natural Security サポートが Object Compare にも適用されます。

このchapterでは、次のトピックについて説明します。

## 開始する前に

---

Object Compare は、Natural スタジオのオプションプラグインです。このプラグインのインストールは Natural のインストールの一部です。このプラグインは初期インストール時にインストールすることも、または後で別のセットアッププロセスとしてインストールすることもできます。インストール手順の詳細については、『インストール』ドキュメントを参照してください。

Object Compare を使用する前に、次のトピックの説明に従う必要があります。

- [前提条件](#)
- [Object Compare の有効化](#)

### 前提条件

ワークファイル 30 が使用されます。したがって、プロファイルパラメータ WORK を 30 以上に設定する必要があります。

ファイル比較用のグラフィカルユーザーインターフェイスを備えた外部ツールが PC にインストールされている必要があります。例えば、ExamDiff などのツールです。diff.exe などの行コマンド比較ツールは機能しません。

### Object Compare の有効化

Object Compare をインストールしたら、プラグインマネージャで有効にする必要があります。有効にするための手順の詳細については、『Natural スタジオの使用』ドキュメントの「プラグインマネージャ」を参照してください。



## Object Compare の要素

プラグインマネージャで Object Compare を有効にすると、Natural スタジオウィンドウで次の追加要素が使用可能になります。

- コンテキストメニュー
- メニューコマンド
- ツールバー
- [オプション] ダイアログボックス

### コンテキストメニュー

現在のコンテキスト（選択したオブジェクト）が対応している場合は、ライブラリワークスペースのコンテキストメニューにカスケードメニュー [Object Compare] が追加されます。このカスケードメニューには、[Write Left File] コマンドと [Write Right File] コマンドがあります。詳細については、「[比較するオブジェクトの選択](#)」を参照してください。


### メニューコマンド

[ツール] > [開発ツール] メニューに追加のコマンド [Start Object Compare] が表示されます。詳細については、「[Object Compare の開始](#)」を参照してください。

### ツールバー

追加のツールバーが表示されます。このツールバーのボタンは、次のメニューコマンドを表します。

#### Object Compare の開始

 **Note:** このツールバーは、[ツール] メニューの [カスタマイズ] を選択して表示と非表示を切り替えることができます。

### [オプション] ダイアログボックス

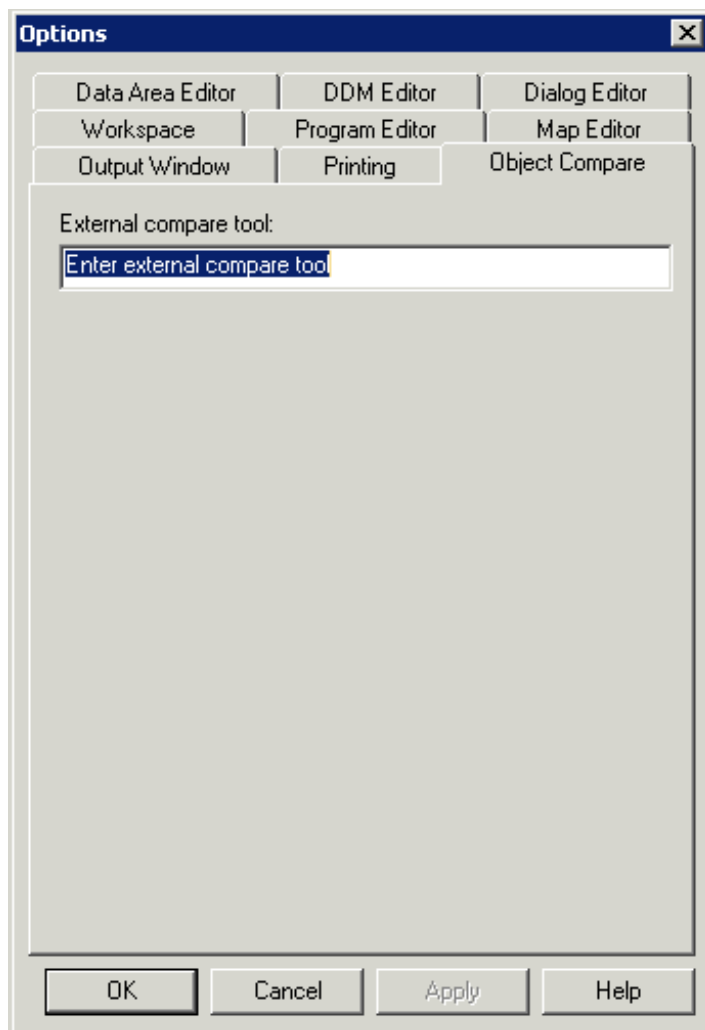
[オプション] ダイアログボックスに、[Object Compare] タブが追加されます。詳細については、「[オプションの設定](#)」を参照してください。

## オプションの設定

Object Compare を有効にしたら、最初にこの Natural スタジオの拡張機能のオプションを設定する必要があります。

### ▶手順 2.1. オプションを設定するには

- 1 [ツール] メニューの [オプション] を選択します。  
[オプション] ダイアログボックスが表示されます。
- 2 このダイアログボックスで [Object Compare] タブを選択します。



- 3 次の情報を指定します。

### External compare tool

ファイルの比較に使用する外部ツール（パスを含む）。

- 4 [OK] ボタンを選択します。

## 比較するオブジェクトの選択

比較するオブジェクトを選択すると、これらのNaturalソースのコピーが、ローカルコンフィグレーションファイルに定義されている一時Naturalディレクトリのサブディレクトリ *left* および *right* に一時的に保存されます。『コンフィグレーションユーティリティ』ドキュメントの「インストール指定」を参照してください。

一時Naturalディレクトリがローカルコンフィグレーションファイルで定義されていない場合は、コピーはWindows環境変数TEMPによって定義されているディレクトリに保存されます。TEMPも定義されていない場合は、コピーはWindows環境変数TMPによって定義されているディレクトリに保存されます。

これらのコピーは外部ツールへの入力です。

### ▶手順 2.2. オブジェクトを選択するには

- 1 ライブラリワークスペースで、比較する1つ目のオブジェクトを選択します。
- 2 コンテキストメニューを表示し、[Object Compare] > [Write Left File] を選択します。  
このオブジェクトのコードが、外部ツールの左側に表示されます。
- 3 ライブラリワークスペースで、比較する2つ目のオブジェクトを選択します。
- 4 コンテキストメニューを表示し、[Object Compare] > [Write Right File] を選択します。  
このオブジェクトのコードが、外部ツールの右側に表示されます。

## Object Compare の開始

Object Compare を開始すると、[オプション] ダイアログボックスで定義した外部ツールが起動し、前の手順で定義した左右のファイルが示されます。

### ▶手順 2.3. Object Compare を開始するには

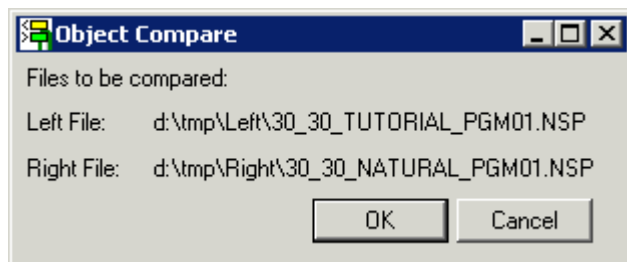
- 1 [ツール] メニューの [開発ツール] > [Start Object Compare] を選択します。

Or:

[Object Compare] ツールバーが表示されている場合は、次のツールバーボタンを選択します。



次のダイアログボックスが表示されます。一時ディレクトリで使用されているファイル名が表示されます。ファイル名は、ホストとポート（リモート開発サーバーのオブジェクトを選択した場合のみ表示）、データベース ID、ファイル番号、ライブラリ名、選択したオブジェクトの名前、および Natural オブジェクトのタイプを示す拡張子で構成されます。



- 2 [OK] ボタンを選択して、外部ツールを起動します。

# 3 ローカルバージョン管理

---

- 予備注釈 ..... 10
- 開始する前に ..... 11
- ローカルバージョン管理の要素 ..... 13
- オプションの設定 ..... 16
- プロジェクトの管理 ..... 20
- ローカルバージョン管理のファーストステップ ..... 27
- ローカルバージョン管理の操作 ..... 44
- ログの処理 ..... 55
- コマンドスクリプト ..... 56

ローカルバージョン管理を使用すると、Natural 環境内部から、ソース制御システム Subversion (SVN) または Concurrent Versions System (CVS) を使用できます。

この機能を使用するには、Subversion または CVS クライアント部分 (*svn.exe* または *cvs.exe*) が PC にインストールされていて、ローカルまたはサーバーにインストールされている Subversion または CVS のリポジトリサーバーにアクセスする必要があります。インターネットを検索すると、複数のダウンロードサイトでこれらの Subversion または CVS コンポーネントを見つけることができます。

このchapterでは、次のトピックについて説明します。

## 予備注釈

---

以下では次のトピックについて説明します。

- [Natural Security](#)
- [サポートされている Subversion および CVS 機能](#)
- [システムライブラリ SYSPLCVS の名前](#)
- [バッチモード](#)

### Natural Security

Natural Security は部分的にサポートされています。ライブラリへのログオンが禁止されている場合は、このライブラリはツリービューに表示されず、ローカルバージョン管理コマンドを実行することはできません。ただし、ライブラリ内の Natural オブジェクトに対して [リスト]、[開く]、または [保存] コマンドを使用できない場合は、Natural オブジェクトを読み取ってリポジトリにコミットしたり、Natural オブジェクトをリポジトリからチェックアウトして Natural にインポート (して置換) したりすることができます。

### サポートされている Subversion および CVS 機能

ローカルバージョン管理では、すべての Subversion および CVS 機能が使用できるわけではありません。したがって、一部のタスクでは、リポジトリのコマンド行インターフェイス (*svn.exe* や *cvs.exe*)、または TortoiseSVN や TortoiseCVS などのフリーサードパーティソフトウェアを使用する必要があります。

## システムライブラリ SYSPLCVS の名前

ローカルバージョン管理のシステムライブラリはSYSPLCVSと呼ばれますが、SubversionとCVS両方のリポジトリのオブジェクトが含まれています。

### バッチモード

現在のバージョンのローカルバージョン管理はオンライン使用を前提に設計されています。バッチモードでリポジトリにアクセスする場合は、オブジェクトハンドラを使用します。

バッチモードでリポジトリから Natural ソースをロードするには、次の手順に従います。

1. システムファイル (FUSER) の構造を持つリポジトリモジュールを、ローカルワークスペースにチェックアウトします。
2. FTOUCHユーティリティを使用して、ライブラリごとにFILEDIR.SAGファイルを作成します。FTOUCHユーティリティ (ストラクチャードモードなど) のパラメータを取得するには、ソースヘッダーを読み取ります (存在する場合)。または、[オプション](#)でソースヘッダーを無効にしている場合は、コミットした各ソースのユーザー定義のデフォルト値を使用できます。

この操作により、システムファイル (FUSER) がワークスペースの外に出され、ローカルで使用したり、バッチモードの Natural でさらに処理したりすることができます。

ソースをリモート開発環境 (NDV) に移動するには、引き続き次の手順に従います。

3. オブジェクトハンドラを使用して、バッチモードでシステムファイル (FUSER) をローカルファイルにアンロードします。
4. バッチモードでリモート開発環境にマッピングします。
5. ダイレクトコマンドを使用して、アンロードされているソースをリモート開発環境にロードします。

詳細については、『ツールおよびユーティリティ』ドキュメントの「オブジェクトハンドラ」の「リモート環境でのバッチ処理」を参照してください。

## 開始する前に

ローカルバージョン管理は、Natural スタジオのオプションプラグインです。このプラグインのインストールは Natural のインストールの一部です。このプラグインは初期インストール時にインストールすることも、または後で別のセットアッププロセスとしてインストールすることもできます。インストール手順の詳細については、『インストール』ドキュメントを参照してください。

ローカルバージョン管理を使用する前に、次のトピックの説明に従う必要があります。

- 前提条件
- ローカルバージョン管理の有効化

### 前提条件

プラグインを有効にすると、Natural スタジオが、パラメータファイル NATPARM を使用して別の Natural セッションを開始します。Natural セッションで NATPARM 以外のパラメータファイルを使用している場合は、FNAT（特にライブラリ SYSPLCVS および SYSLIBS）や他の Natural パラメータ（WORK パラメータなど）が Natural プラグインセッション（NATPARM を使用）と Natural セッション（別のパラメータファイルを使用）が一致しないことがある点に注意してください。このために予測できないエラーが発生する場合があります。このため、Natural セッションでパラメータファイル NATPARM も使用することをお勧めします。Natural パラメータファイルの詳細については、『*Natural Security*』ドキュメントの「*Natural Security* でのプラグイン」を参照してください。

ワークファイル 29、30、31、および 32 が使用されます。したがって、パラメータファイル NATPARM でプロファイルパラメータ WORK を 32 に設定する必要があります。

ローカルバージョン管理では、Subversion のコマンドインターフェイス *svn.exe* または CVS のコマンドインターフェイス *cvs.exe*（厳密にこれらの名前です）を使用します。これらは、ローカルバージョン管理を実行する PC にインストールされている必要があります。コマンドインターフェイスがアクセス可能であることを確認してください（例えば、Windows 環境変数 %PATH% に *svn.exe* または *cvs.exe* をパスを含めて追加します）。

Windows レジストリで *cmd.exe* のコマンド拡張機能が有効になっている必要があります。これはデフォルトで有効になっています。

ソース制御システムのリポジトリにアクセスできる権限が必要です。このリポジトリは、ローカルまたはサーバーにインストールできます。ローカルバージョン管理を使用する前に、リポジトリが動作している必要があります。

Natural オブジェクトをアーカイブする際は、行番号または行番号参照を含めないでください。行番号は、Natural オブジェクトがワークスペースにコピーされるときにローカルバージョン管理によって削除されます。行番号が削除されるのは、ソースで 1 行を追加または削除したときに行番号のずれが差異にならないようにするためです。ワークスペースとして使用されるライブラリでは番号の省略をオンにすることを強くお勧めします。『*Natural スタジオの使用*』ドキュメントの「ソースコードの行番号の省略」を参照してください。

ローカルバージョン管理では、ファイルシステム名をそのまま使用して Natural オブジェクトをリポジトリに保存します。Natural オブジェクトの名前に特殊文字が含まれている場合は、Natural 名とファイルシステム名が異なる名前になります。特殊文字を含む Natural オブジェクトをリモートメインフレーム開発サーバー（NDV）から Natural 外部のワークスペースに移動した場合は、ファイルシステム名を使用できません。ローカルバージョン管理では、このような Natural オブジェクトはサポートされません。Natural 名に特殊文字を使用しないことを強くお勧めします。



特殊文字は次のとおりです。

■ リモート **UNIX** または **OpenVMS** 開発サーバーの場合

.  
#  
@  
すべて小文字 (a~z)

■ リモートメインフレーム開発サーバーの場合

#  
\$  
%  
+  
/  
@  
§  
すべて小文字 (a~z)

## ローカルバージョン管理の有効化

ローカルバージョン管理をインストールしたら、プラグインマネージャで有効にする必要があります。有効にするための手順の詳細については、『*Natural スタジオの使用*』ドキュメントの「プラグインマネージャ」を参照してください。

## ローカルバージョン管理の要素

プラグインマネージャでローカルバージョン管理を有効にすると、Natural スタジオウィンドウで次の追加要素が使用可能になります。

- メニューコマンド
- ツールバー
- コンテキストメニュー

### ■ [オプション] ダイアログボックス

#### メニューコマンド

[オブジェクト] メニューにカスケードメニュー **[Local Versioning]** が表示され、オブジェクトを処理するコマンドを使用できます。これらのコマンドは、**コンテキストメニュー**にも表示されます（以下を参照）。コンテキストメニューを使用できないNaturalオブジェクト（リソースやエラーメッセージなど）の場合は、[オブジェクト] メニューからこれらのコマンドを使用できます。


[ライブラリ] メニューにカスケードメニュー **[Local Versioning]** が表示され、ライブラリを処理するコマンドを使用できます。これらのコマンドは、**コンテキストメニュー**にも表示されます（以下を参照）。これらのコマンドを使用して、ソース制御システムへの新しいライブラリの追加などを行うことができます。また、Naturalに新しいライブラリを作成して、その空のライブラリに対してコマンドを実行できます。

[ツール] > [開発ツール] メニューにカスケードメニュー **[Local Versioning]** が表示され、**[Maintain Projects]** コマンドを使用できます。このコマンドは、ローカルバージョン管理のプロジェクトの定義と管理に使用します。

#### ツールバー


追加のツールバーが表示されます。このツールバーのボタンは、次のメニューコマンドを表します。

#### **Maintain Projects**

 **Note:** このツールバーは、[ツール] メニューの [カスタマイズ] を選択して表示と非表示を切り替えることができます。

#### コンテキストメニュー

現在のコンテキスト（選択したオブジェクト）が対応している場合は、コンテキストメニューにカスケードメニュー **[Local Versioning]** が追加されます。

 **Note:** ローカルバージョン管理コマンドは Natural リソースと Natural エラーメッセージに適用できませんが、コンテキストメニューにカスケードメニュー **[Local Versioning]** は表示されません。この場合は、代わりに**メニューコマンド**を使用してください。

カスケードメニューに含まれるコマンドは、コンテキストに応じて変化します。

**プロジェクト定義**によっては、プロジェクトフォルダのステータス、**オプション** **[Disable import, make branch, add library]** の設定、およびカスケードメニューのいくつかのコマンドが無効になることがあります。

## システムファイルレベル

システムファイルを選択しているときは（FUSER または FNAT のノードなど）、カスケードメニューに次のコマンドが表示されます。

Subversion のコマンド	CVS のコマンド
SVN Display Log	CVS Display Log
SVN Status System File	CVS Status System File
SVN Import Module into Repository	CVS Import Module into Repository
SVN Make Branch	CVS Make Branch/Tag
SVN Commit System File	CVS Commit System File
SVN Update System File	CVS Update System File
SVN Check Out and Import System File	CVS Check Out and Import System File
SVN Check Out System File	CVS Check Out System File
Import System File into Natural	Import System File into Natural

## ライブラリレベル

ライブラリを選択しているときは、カスケードメニューに次のコマンドが表示されます。

Subversion のコマンド	CVS のコマンド
SVN Display Log	CVS Display Log
SVN Status Library	CVS Status Library
SVN Add and Commit Library	使用できるコマンドはありません。
SVN Add Library	CVS Add Library
SVN Delete and Commit Library	CVS Remove and Commit Library
SVN Delete Library	CVS Remove Library
SVN Commit Library	CVS Commit Library
SVN Update Library	CVS Update Library
SVN Check Out and Import Library	CVS Check Out and Import Library
SVN Check Out Library	CVS Check Out Library
Import Library into Natural	Import Library into Natural

### Natural オブジェクトレベル

Natural オブジェクトを選択しているときは、カスケードメニューに次のコマンドが表示されます。

Subversion のコマンド	CVS のコマンド
SVN Display Log	CVS Display Log
Copy and SVN Commit Object	Copy and CVS Commit Object
Copy to Project Folder	Copy to Project Folder
SVN Add and Commit Object	CVS Add and Commit Object
SVN Add Object	CVS Add Object
SVN Delete and Commit Object	CVS Remove and Commit Object
SVN Delete Object	CVS Remove Object
SVN Commit Object	CVS Commit Object
SVN Update Object	CVS Update Object
使用できるコマンドはありません。	CVS Check Out and Import Object
使用できるコマンドはありません。	CVS Check Out Object
Import Object into Natural	Import Object into Natural

上記のコマンドは、Subversion および CVS によって使用されるコマンドに対応しています。

#### [オプション] ダイアログボックス

[オプション] ダイアログボックスに、[**Local Versioning**] タブが追加されます。詳細については、「[オプションの設定](#)」を参照してください。

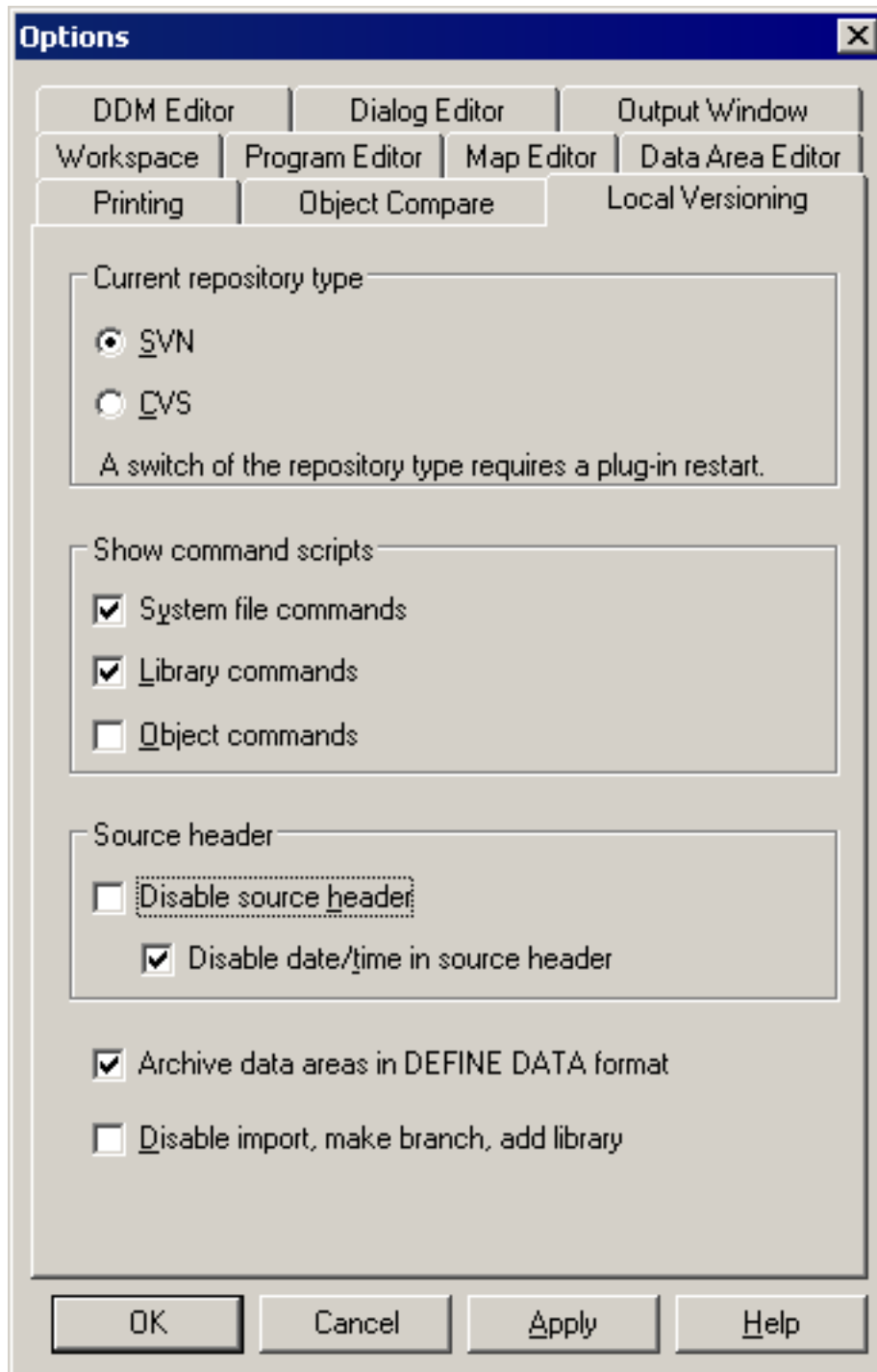
## オプションの設定

---

ローカルバージョン管理を有効にすると、この Natural スタジオの拡張機能のオプションを設定できるようになります。

#### ▶手順 3.1. オプションを設定するには

- 1 [ツール] メニューの [オプション] を選択します。  
[オプション] ダイアログボックスが表示されます。
- 2 このダイアログボックスで [**Local Versioning**] タブを選択します。



- 3 次のオプションを設定できます。

#### Current repository type

Natural オブジェクトソースをコミットしたりチェックアウトしたりするリポジトリタイプのオプションボタン [SVN] (Subversionの場合) または [CVS] を選択します。

リポジトリタイプを切り替えた場合は、プラグインマネージャでローカルバージョン管理を再起動する必要があります（いったん無効化され、有効化されます）。詳細については、『Natural スタジオの使用』ドキュメントの「プラグインマネージャ」を参照してください。

### Show command scripts

このグループボックスのチェックボックスをオンにすると、ローカルバージョン管理に実装されている Subversion または CVS コマンドを発行するたびに、Windows コマンドプロンプトのウィンドウが表示されます。これにより、Subversion または CVS コマンドが成功したかどうかを確認できます。

[**System file commands**] はシステムファイルレベルの、[**Library commands**] はライブラリレベルの、[**Object commands**] は Natural オブジェクトレベルのウィンドウを有効化/無効化します。

上記のコマンドのウィンドウを無効にしている場合は、ローカルバージョン管理の別のコマンドを使用してステータスを表示できます。または、TortoiseSVN や TortoiseCVS などのサードパーティツールを使用して、コマンドが成功したかどうかを確認できます。

初めて使用する場合やテスト目的の場合は、すべてのレベルを有効にできます。ライブラリ全体のオブジェクトなど、一度に多数のオブジェクトを選択した場合は、オブジェクトレベルのウィンドウの数が膨大になるため、オブジェクトレベルをオフにすることをお勧めします。選択したオブジェクトの数が 10 を超えた場合は、警告が表示されます。

選択したオブジェクトの処理は、Esc キーを押してキャンセルできます。ただし、このグループボックスのオプションをオンにした状態で、コマンドウィンドウが多数ある場合は、Natural が制御を取得するのが難しくなります。

例外：**ライブラリまたはシステムファイルのステータス**を表示するコマンドを発行したとき、または**プロジェクト定義**を含むダイアログボックスで [**Login**] ボタンを選択したときは、必ずコマンドプロンプトウィンドウが表示されます。これはオプションの設定に関係ありません。

### Source header

ローカルバージョン管理では、Natural ソースをワークスペースにコピーする前に、ソースにヘッダーを追加します。ヘッダーには、オブジェクトの Natural 名、ソースユーザー ID、モード、コードページ、ソースの日付、およびソースの時刻が含まれます。この情報は、リポジトリからチェックアウトしたソースオブジェクトを Natural にインポートするときに使用されます。

ローカルバージョン管理がソースをインポートする前にソースのヘッダーを見つけられなかった場合は、デフォルト値が使用されます。デフォルト値は次のとおりです。

ヘッダータイプ	デフォルト値
Mode	Structured
オブジェクトの Natural 名	ワークスペース/リポジトリ内のファイルの名前
ユーザー ID	*USER
コードページ	*CODEPAGE
ソースの日付	*DATN
ソースの時刻	*TIMN

ソースに変更がない場合でもソースの日時が変更されることがあるため、他のヘッダーパラメータはそのまま、ソースの日付とソースの時刻を無効にすることができます。このグループボックスでは、次のチェックボックスを使用できます。

#### ■ Disable source header

このチェックボックスをオンにすると、ソースの日付とソースの時刻を含むソースヘッダーすべてが無効になります。

#### ■ Disable date/time in source header

このチェックボックスをオンにすると、ヘッダーからソースの日付とソースの時刻が除外されます。その他のパラメータは、**[Disable source header]** チェックボックスの設定に依存します。

ローカルバージョン管理では、Natural for Eclipse のヘッダーがある Natural ソースを処理できます。

#### Archive data areas in DEFINE DATA format

このチェックボックスをオンにすると、データエリアをワークスペースにコピーしてリポジトリにコミットする前に、データエリアが DEFINE DATA フォーマットに変換されます。

このチェックボックスをオフにすると、データエリアはファイルシステム内のデータエリア用の Natural 内部フォーマットでアーカイブされます。

DEFINE DATA フォーマットは読み取り可能です。内部フォーマットのデータエリアを FTOUCH ユーティリティまたは Natural のインポート機能を使用して Natural にインポートするときに変換する必要はありません。

データエリアに含まれるビューにマルチプルフィールドまたはピリオディックグループとしてマークされたフィールドがある場合（ドキュメントの場合のみ。ビューを使用する場合は関係ありません）、そのデータエリアを DEFINE DATA フォーマットに変換すると、マルチプルフィールドまたはピリオディックグループに関する情報は失われます。

## Disable import, make branch, add library

このチェックボックスをオンにすると、次のコマンドが無効になります。

Subversion のコマンド	CVS のコマンド
SVN Import Module into Repository	CVS Import Module into Repository
SVN Make Branch	CVS Make Branch/Tag
SVN Add Library	CVS Add Library
SVN Add and Commit Library	使用できるコマンドはありません。

4 [OK] ボタンを選択します。

## プロジェクトの管理

以下では次のトピックについて説明します。

- プロジェクトのリストの表示
- 新規プロジェクトの定義
- プロジェクトの削除

### プロジェクトのリストの表示

ローカルバージョン管理を使用する前に、プロジェクトを定義する必要があります。各プロジェクトは特定のシステムファイルに属し、環境、データベースID (DBID)、およびファイル番号 (FNR) で識別されます。

Naturalシステムファイルは開発環境です。この中でプロジェクトのソースの作成、変更、およびテストを行います。通常、このようなNaturalシステムファイルはローカル環境またはリモート開発サーバーにある FUSER です。ローカルバージョン管理はすべてのタイプの Natural システムファイルをサポートしているため、FNAT や非アクティブ環境にも対応しています。非アクティブ環境とは、コンフィグレーションユーティリティのグローバルコンフィグレーションファイルでシステムファイルとして定義されたファイルです。このファイルには、Natural スタジオのライブラリワークスペースからアクセスできます。ソースファイルは、非アクティブ環境で変更することはできませんが、非アクティブ環境にコピーすることは可能です。

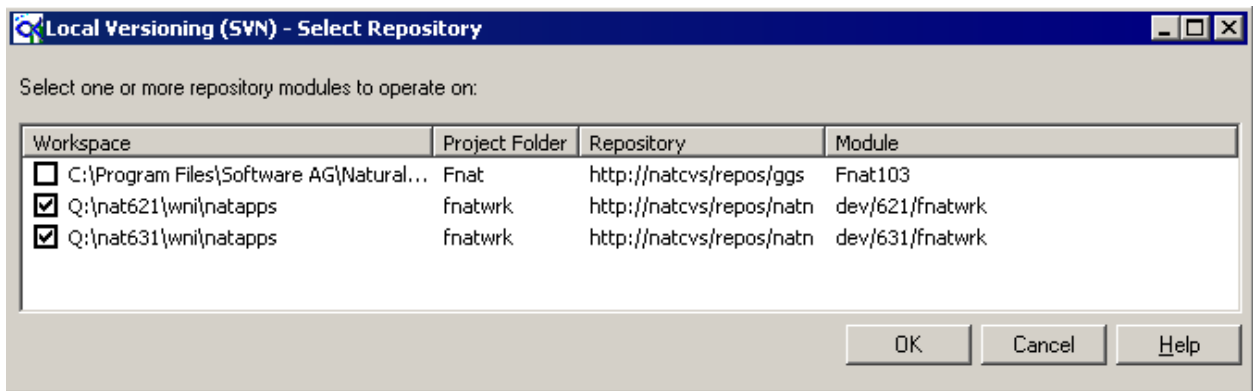
ワークスペースは、Natural 外部の通常の Windows フォルダにあります。また、Natural システムファイルである場合もあります。非アクティブな Natural for Windows システムファイルをリポジトリワークスペースに使用すると、ユーザーが Natural ライブラリワークスペース内でリポジトリワークスペースを参照できるという利点があります。また、アクティブシステムファイル (FUSER など) を、ユーザーの開発ソース保持場所とリポジトリワークスペースの両方の目的で使用することもできます。この場合は、ユーザーはソースが FUSER のものなのかリポジトリワークスペースのものなのかに注意する必要があります。FUSER内のソースには、通常、ヘッ



ダーはなく、行番号と行番号参照が含まれていることがあります。ワークスペース内のソースには、通常、ヘッダーはありますが、行番号と行番号参照はありません。

各ワークスペースは Subversion または CVS リポジトリを 1 つだけ参照できます。リポジトリは複数のワークスペースを参照できます。

1 つの Natural システムファイルに、ワークスペースを含む複数のリポジトリを定義できます。その場合は、次のダイアログボックスが表示されます。システムファイル内の Natural オブジェクトに対してコマンドを実行するときに、ワークスペースを含む 1 つまたは複数のリポジトリを選択する必要があります。



ワークスペースを含む 1 つのリポジトリに複数の Natural システムファイルを定義することもできます。

プロジェクトを設定するときに [Maintain Projects] コマンドを使用した場合は、リポジトリ内の Natural ライブラリまたはオブジェクトの混在や上書きを防ぐための警告が表示されます。このような場合は、オブジェクトの上書きや間違ったライブラリまたはシステムファイルへのコミットまたはチェックアウトを不用意に行わないように注意する必要があります。

### ▶手順 3.2. プロジェクトのリストを表示するには

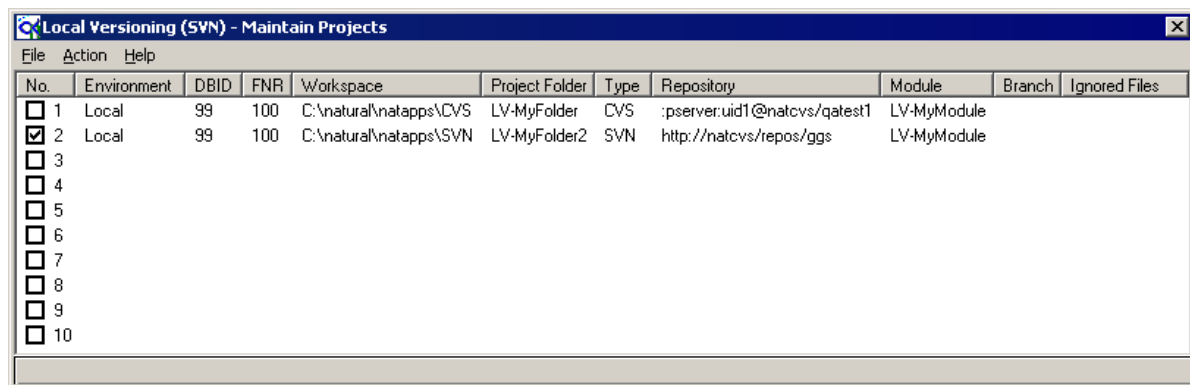
- [ツール] メニューの [開発ツール] > [Local Versioning] > [Maintain Projects] を選択します。

Or:

[Local Versioning] ツールバーが表示されている場合は、次のツールバーボタンを選択します。



次のウィンドウが開き、プロジェクトリストが表示されます。最大10件のプロジェクトを定義できます。最初の列にあるチェックボックスを使用して、プロジェクトを有効化/無効化できます。初期状態では、プロジェクト定義は空です。現在使用されているリポジトリタイプは、タイトルバーに示されます。例えば、次のようになります。



**Note:** 定義済みプロジェクトのリストは、システムライブラリ SYSPLCVS にリソース（テキストドキュメント）として保存されます。このファイルの名前は *lv\_projects.txt* です。このファイルをは削除しないでください。削除すると、すべてのプロジェクト定義が失われます。プラグインのインストール時には、また Natural の新規インストール時でも、以前の *lv\_projects.txt* ファイルは維持されます。プラグインを有効にするときに *lv\_projects.txt* ファイルが存在しない場合は、ファイルがデフォルト値で作成されます。

## 新規プロジェクトの定義

プロジェクトを定義するには、プロジェクトリストから空のエントリを1つ選択し、プロジェクト定義を編集します。

### ▶手順 3.3. プロジェクト定義を編集するには

- 1 プロジェクトを選択し、[Action] メニューの [Maintain] を選択します。

Or:

プロジェクトを選択し、コンテキストメニューを表示し、[Maintain] を選択します。

Or:

プロジェクトをダブルクリックします。

次のダイアログボックスが表示されます。

Local Versioning (SVN) - Maintain Project 1

Natural

Environment name:

DBID:

ENR:

Ignored files:

Library Mapping

Natural Library	Repository Library

New Edit Delete

Local

Workspace:

Project folder:

Repository

Type:

Path:  Login

Module:

Branch:

OK Cancel Apply Help

2 次の情報を指定します。

**Environment name**

ドロップダウンリストボックスから、ソース制御システムに追加する Natural ソースがある Natural システムファイルの環境（ローカルまたはリモート開発サーバー）を選択します。

**DBID**

ソース制御システムに追加する Natural ソースがある Natural システムファイルのデータベース ID です。

**FNR**

ソース制御システムに追加する Natural ソースがある Natural システムファイルのファイル番号です。

**Ignored files**

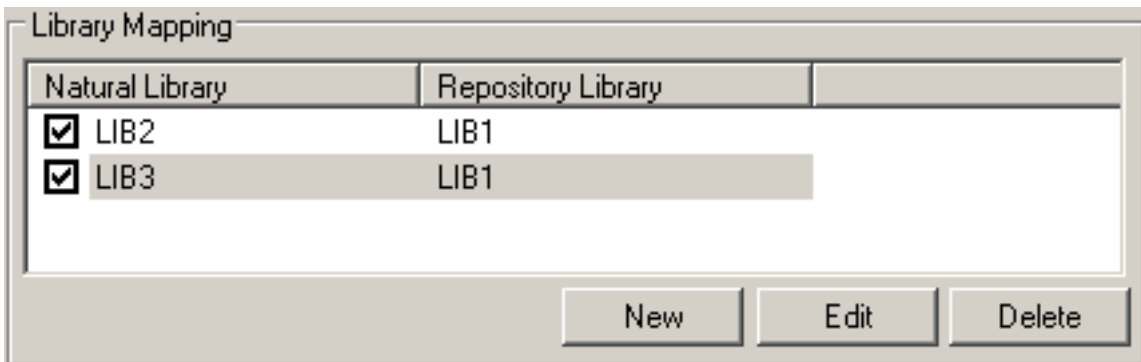
オプション。無視される Natural ソースを指定できます（選択されていても無視されます）。無視された Natural ソースは、ローカルバージョン管理で処理されません。

Natural ソース名とこのフィールドの入力では、大文字と小文字が区別されます。Natural オブジェクトの名前に含まれるアンパサンド（&）が言語コードで置換され、オブジェクト名に小文字が含まれることがあります。ただし、ほとんどの場合は Natural ソース名は大文字です。

ソース名はセミコロン（;）で区切ります。ファイル拡張子は指定できません。ワイルドカードは使用できません。

**Library Mapping**

オプション。オブジェクトのコミットまたはチェックアウトを目的として Natural ライブラリを変更する場合は、**[New]** ボタンを使用して、ライブラリを指定し、ソース制御システムの他のライブラリにマッピングします。ライブラリマッピングは有効化/無効化できます。例えば、次のようになります。



## Workspace

Naturalオブジェクトをソース制御システムにコピーする前に保持するローカル作業フォルダへのパスです。通常、ワークスペースはFNATやFUSERなどのNaturalシステムファイルと並行させます（Natural外部や非アクティブNatural環境などで）。ただし、アクティブシステムファイル（FUSERなど）を指定することも可能です。次に例を示します。

```
c:\natural\cvs
```

UNCパス（\\nat\cvsなど）はサポートされません。cmd.exeがUNCパスをサポートしていないためです。



**Note:** パフォーマンス上の理由で、Naturalの外部にワークスペースを定義できません。Natural外部のワークスペースで作業する方が、Naturalとの対話が不要なため高速になります。

## Project folder

ワークスペース内のサブフォルダです。

ソース制御システムのモジュールごととブランチごとに独立したサブフォルダを定義することをお勧めします。



### Notes:

1. 非アクティブシステムファイルをワークスペース/プロジェクトフォルダとして使用すると、Natural環境内でワークスペースを参照できるという利点があります。
2. ワークスペースがNaturalシステムファイルの場合は、SubversionまたはCVSステータス情報用のフォルダ（.svnまたはCVSが付いた名前）がリポジトリによってNaturalシステムファイル内に作成されます。CVSを使用している場合は、FUSERにCVSという名前のライブラリがあると、問題が発生することがあります。この名前のライブラリがある場合は、CVSでは管理ファイルがCVSフォルダに配置されます。Naturalライブラリ全体を削除したときに、Naturalが認識できないファイルまたはフォルダ（.svnやCVSなど）は削除されません。
3. Naturalソースをプロジェクトフォルダにコピーするのに [SVN Add Object]、[CVS Add Object]、または [Copy to Project Folder] コマンドを使用した場合は、行番号がソースから削除され（ソースで1行を追加または削除したときに行番号のずれが差異にならないようにするため）、行番号参照が記号ラベルで解決されます。FUSERをプロジェクトフォルダとして定義した場合は、行番号を削除するためにこれらのコマンドを実行する必要があります。
4. SubversionまたはCVSディレクトリからFUSERにNaturalソースをチェックアウトしても、このソースがFUSER上にまだ存在していない場合は、このソースをNaturalで参照することはできません。これは、ファイルFILEDIR.SAGが更新されていないためです。ソースをNaturalにインポートする必要があります。

5. 指定したワークスペース／プロジェクトフォルダディレクトリがない場合は、最初に使用するときには作成されます。

### Type

リポジトリタイプは [SVN] (Subversion の場合) または [CVS] です。

### Path

ソース制御システムへの完全な接続文字列です。

#### ■ Subversion

Subversion の場合は、この文字列は次のいずれかのフォーマットで入力できます (Subversion の構成による)。

```
file://d:...
```

```
svn://...
```

```
http://...
```

例えば、次のようになります。

```
http://natsvn/repos/proj1/dev/vers/fuser100
```



**Note:** Subversion の場合は、リポジトリからのチェックアウトおよびリポジトリへのインポートのコマンドにのみこのパスが使用されます。その他の Subversion コマンドでは、`.svn` ディレクトリのパス指定が使用されます。

#### ■ CVS

CVS の場合は、この文字列は次のフォーマットで入力する必要があります。

```
:protocol:user-name@server-name:repository-folder
```

例えば、次のようになります。

```
:pserver:natural@cvs.myserver.com:/cvsroot/Repositories/DevSources
```



**Tip:** 文字列をソース制御システムから直接コピーし、このテキストボックスに貼り付けます。

### Login

このボタンを選択すると、リポジトリパスワードを入力するためのコマンドプロンプトウィンドウが表示されます。これは、`svn info` または `cvs login` コマンドに対応しています。これらのコマンドは、パスワードの確認と、パスワードを今後使用できるようにホームディレクトリに保存するのに使用します。

**Module**

使用／作成する Subversion または CVS モジュールの名前です。

**Branch**

オプション。

Subversion の場合は、メインバージョンのブランチを保存する新しいディレクトリの名前です。

CVS の場合は、作業対象のブランチまたは作成するブランチの名前です。

ブランチを指定しない場合は、メインブランチが使用されます。

3 [OK] ボタンを選択します。

## プロジェクトの削除

[Maintain Projects] ウィンドウで現在定義されている任意のプロジェクトを削除できます。

### ▶手順 3.4. プロジェクトを削除するには

■ プロジェクトを選択し、[Action] メニューの [Delete] を選択します。

Or:

プロジェクトを選択し、コンテキストメニューを表示し、[Delete] を選択します。

それまでにプロジェクト定義に指定されていたすべての情報が削除されます。

また、システムライブラリ SYSPLCVS にリソース（テキストドキュメント）として保存される *lv\_projects.txt* ファイル内でプロジェクトを削除することもできます。

## ローカルバージョン管理のファーストステップ

このセクションでは、簡単なチュートリアルによって、ソース制御システムの操作方法を示します。チュートリアルは Subversion と CVS の両方に対応しています。

次のトピックについて説明します。

- [このチュートリアル的前提条件](#)
- [ソース制御システムにおける新規プロジェクトの設定](#)
- [ソース制御システムのプロジェクトのブランチ作成](#)
- [メインブランチでの作業](#)

- 別のブランチでの作業

このチュートリアル的前提条件

プロジェクト定義

このチュートリアルでは、プロジェクト定義に次の値が指定されていると仮定します。

Subversion の場合：



**Local Versioning (SVN) - Maintain Project 1**

**Natural**

Environment name: Local

DBID: 30

ENR: 30

Ignored files:

**Library Mapping**

Natural Library	Repository Library
-----------------	--------------------

New Edit Delete

**Local**

Workspace: C:\natural\natapps\SVN

Project folder: LV-MyFolder

**Repository**

Type: SVN

Path: http://natsvn/repos/demo Login

Module: LV-MyModule

Branch:

OK Cancel Apply Help

CVS の場合：

The screenshot shows a dialog box titled "Local Versioning (CVS) - Maintain Project 1". It contains the following fields and sections:

- Natural**
  - Environment name: Local
  - DBID: 30
  - FNR: 30
  - Ignored files: (empty text box)
- Library Mapping**

Natural Library	Repository Library

Buttons: New, Edit, Delete
- Local**
  - Workspace: C:\natural\natapps\CVS
  - Project folder: LV-MyFolder
- Repository**
  - Type: CVS
  - Path: :pserver:uid1@natcvs/qatest1 (with Login button)
  - Module: LV-MyModule
  - Branch: (empty text box)

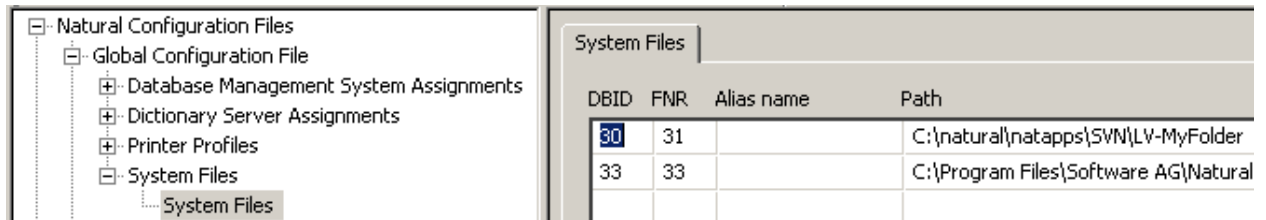
Buttons at the bottom: OK, Cancel, Apply, Help

Natural 環境（環境名、DBID、FNR）とリポジトリ（リポジトリタイプとパス）の正確な値が指定されていることを確認してください。ワークスペース、プロジェクトフォルダ、およびモ

ジュール名については、上記の値を使用します。残りのテキストボックスは空白のままにしておきます。

## ワークスペース定義

コンフィグレーションユーティリティを使用して、ワークスペースを非アクティブ環境として定義します（以下の例では、DBID 30 と FNR 31 は Subversion のワークスペース内のプロジェクトフォルダ `C:\natural\natapps\SVN\LV-MyFolder` をポイントしています）。



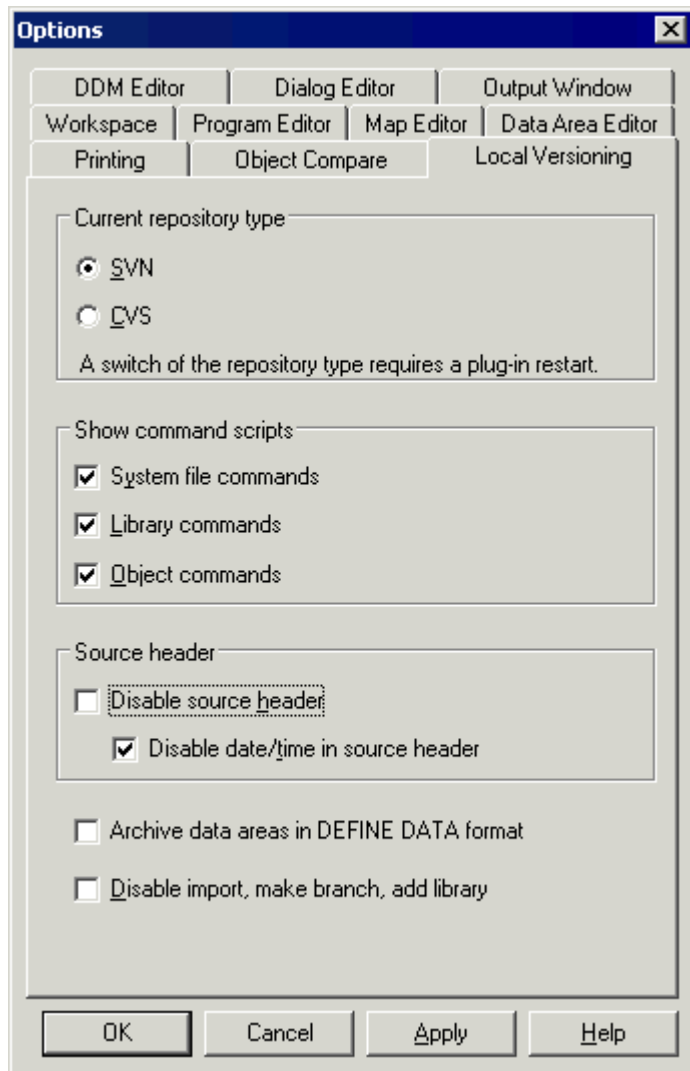
この非アクティブ環境が Natural スタジオのライブラリワークスペースに表示されます。



## オプション

このチュートリアルでは、次のオプションが有効になっていると仮定します。

Subversion の場合：




CVSの場合：リポジトリタイプとして [CVS] を指定する以外は同じです。

### ライブラリとオブジェクト

このチュートリアルでは、LV-TEST という名前のライブラリが存在し、このライブラリに PGM1 という名前のプログラムと SUBPGM1 という名前のサブプログラムが含まれていると仮定します。

## ソース制御システムにおける新規プロジェクトの設定

この演習では、新規モジュールに新規プロジェクトを設定する方法と、Naturalライブラリ全体をソース制御システムに追加およびコミットする方法を学習します。

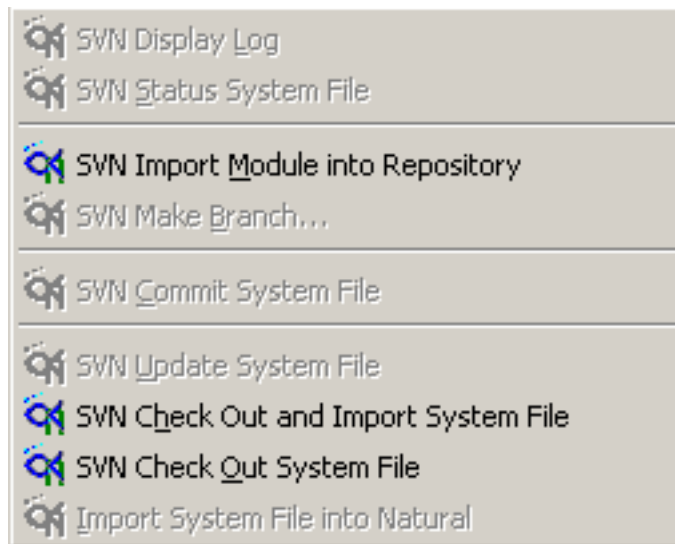
 **Note:** このチュートリアルの説明は、ライブラリワークスペースの論理ビューで作業する場合に適用されます。

### ▶手順 3.5. ソース制御システムに新規プロジェクトを設定するには

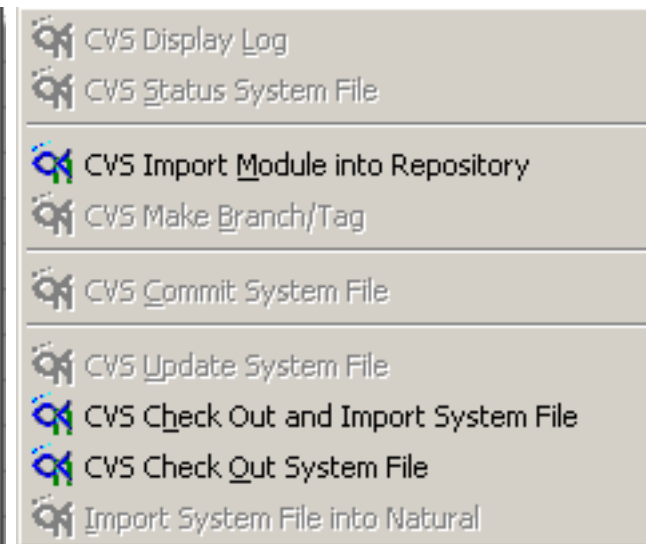
- 1 Natural ソースが含まれるシステムファイルのノード ("User Libraries" など) を選択します。
- 2 選択したノードのコンテキストメニューを開き、カスケードメニュー [Local Versioning] のコマンドを表示します。

以下のコマンドが表示されます。

Subversion の場合：




CVS の場合：



プロジェクト定義でブランチが指定されていないため、ブランチ/タグを作成するコマンドは無効になっています。

Subversion または CVS バージョン制御下にまだワークスペースが存在しないため、それ以上のコマンドも無効になっています。

 **Note:** 上記のコマンドがすべて無効になっている場合は、選択したシステムファイルとプロジェクト定義の指定が一致していません。

- 3 新規モジュールに新規プロジェクトを設定するには、コンテキストメニューのコマンド [SVN Import Module into Repository] または [CVS Import Module into Repository] を選択します。

ダイアログボックスが表示されるので、インポートコマンドのログメッセージを入力します。プリセットのログメッセージは変更しないでください。

- 4 次に進むには [OK] ボタンを選択します。

[Show command scripts] グループボックスのオプションが有効になっているため、コマンドプロンプトウィンドウが表示され、ソース制御システムによって実行されたコマンドおよびその実行結果が示されます。

- 5 Enter キーを押して次に進みます。

この時点でリポジトリをチェックすると (CVS と Subversion の両方に対応したブラウザで ViewCVS を起動するなど)、プロジェクト定義に指定したリポジトリにモジュール LV-MyModule が作成されていることを確認できます。

ローカルワークスペースをチェックすると (Windows エクスプローラを使用するなど)、フォルダ LV-MyFolder が作成されていることを確認できます。このフォルダには .svn または CVS という名前のフォルダが含まれています。このフォルダは、新規モジュール LV-MyModule の明示的なチェックアウトによって作成されています。

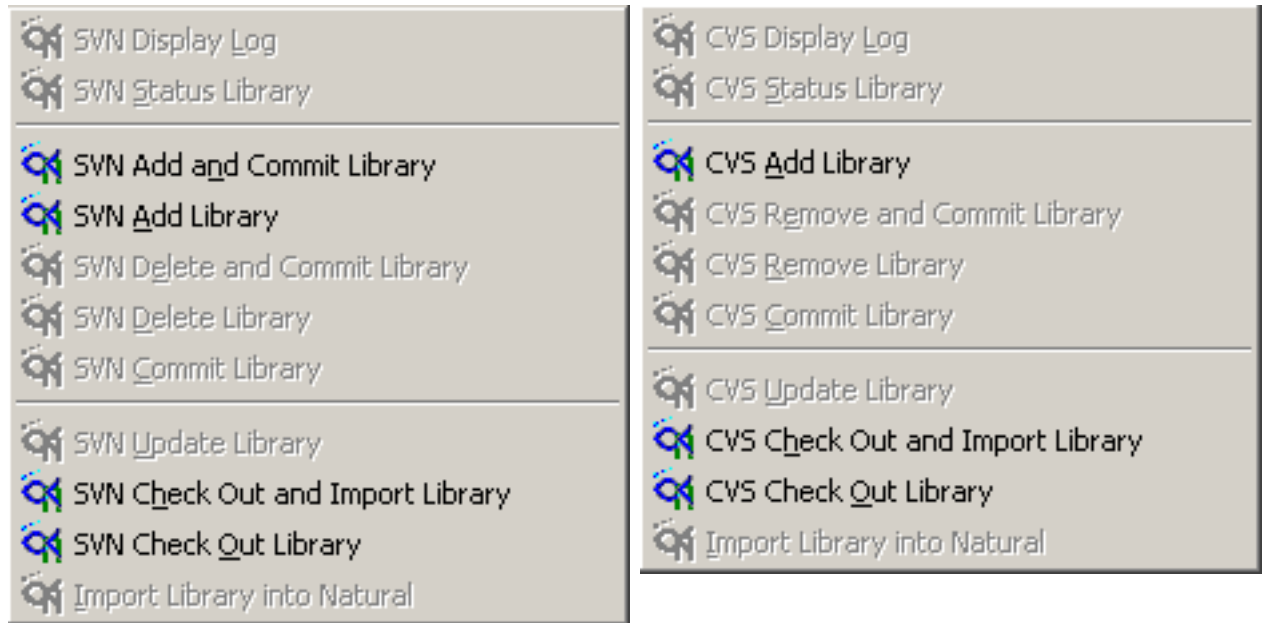
この時点でシステムファイルにプロジェクトが設定されているため、ソース制御システムにライブラリを追加できます。

- 6 ライブラリのノードを選択し (このチュートリアルではこのフォルダの名前が LV-TEST であると仮定します)、コンテキストメニューを開き、カスケードメニュー [Local Versioning] のコマンドを表示します。

以下のコマンドが表示されます。

Subversion の場合：


CVS の場合：



選択したライブラリはまだワークスペース内のバージョン制御下にないため（オブジェクトの追加やチェックアウトが行われていないため）、ライブラリの追加とチェックアウトのコマンドのみが有効になっています。このコンテキストメニューのその他すべてのコマンドは無効になっています。

- 7 選択したライブラリをソース制御システムに追加するには、コンテキストメニューのコマンド **[SVN Add Library]** または **[CVS Add Library]** を選択します。

Subversion：次にコマンド **[SVN Commit Library]** を選択してライブラリをコミットします。

 **Note:** コマンドを実行してコマンドプロンプトウィンドウが表示されたら、前述の説明に従って手順を進めます。このチュートリアルでは対応する説明は行いません。

Natural スタジオのライブラリワークスペースでは、この時点で、ワークスペースとして定義した非アクティブシステムファイル (30,31) にライブラリ LV-TEST も作成されていることを確認できます。



この時点でリポジトリをチェックすると（CVS と Subversion の両方に対応したブラウザで ViewCVS を起動するなど）、フォルダ LV-TEST（同名の Natural ライブラリに対応）がモジュール LV-MyModule に作成されていることを確認できます。

ローカルプロジェクトフォルダ LV-MyFolder をチェックすると（Windows エクスプローラを使用するなど）、フォルダ LV-TEST が作成されていることを確認できます。このフォル

ダには、Subversion ステータス情報を含む *.svn* という名前のフォルダか、CVS ステータス情報を含む CVS という名前のフォルダが含まれています。

Natural ライブラリに対応するディレクトリが作成されているため（Subversion または CVS のステータスファイルを含む）、この時点でこのライブラリのコンテキストメニューのコマンドがすべて有効になります。

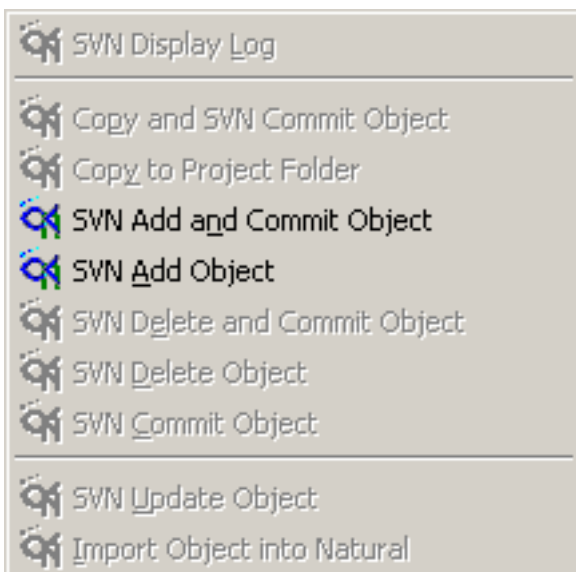


**Note:** CVS のみ：オブジェクトとは異なり、ディレクトリを CVS にコミットする必要はありません。ディレクトリは、コミットしなくても CVS に作成されます。

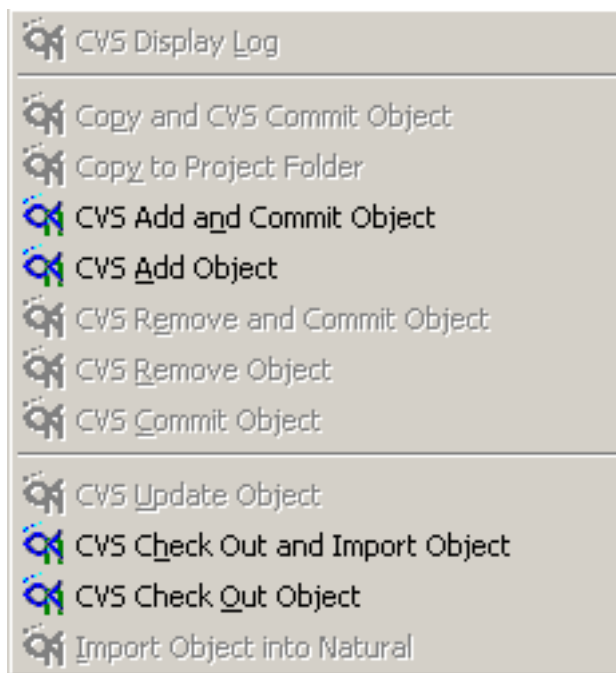
- ライブラリ内のオブジェクトのノードを選択し（このチュートリアルではこのオブジェクトの名前が PGM1 であると仮定します）、コンテキストメニューを開き、カスケードメニュー **[Local Versioning]** のコマンドを表示します。

以下のコマンドが表示されます。

Subversion の場合：



CVS の場合：



選択したオブジェクトはまだソース制御システムに追加されていないため（バージョン制御下にないため）、CVS へのオブジェクトの追加と CVS からのオブジェクトのチェックアウトのコマンドのみが有効になっています。



**Notes:**

- コマンド **[CVS Check Out Object]** は、オブジェクトがソース制御システムにすでに保存されている場合のみオブジェクトを返すことができます。オブジェクト PGM1 がすでに



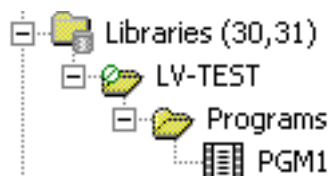
- ソース制御システム（*LV-MyModule\LV-TEST\SRC* 内）に保存されていれば、このコマンドでオブジェクトを読み取ることができます。
2. Subversion を使用している場合は、アプリケーションの保全性を確保するためにライブラリ全体をチェックアウトできます。
- 9 コンテキストメニューのコマンド **[SVN Add Object]** または **[CVS Add Object]** を選択します。

サブフォルダ *SRC* がまだ存在しないため、自動的に作成され、ソース制御システムにコミットされます。

オブジェクトはプロジェクトフォルダ（*PGM1* が非アクティブ環境に表示されています）にコピーされ、同時にソース制御システムにコピーする準備が行われます。

ローカルプロジェクトフォルダで *SRC* フォルダをチェックすると（Windows エクスプローラを使用するなど）、ファイル *PGM1.NSP* が作成されていることを確認できます。これはファイルシステム内の *PGM1.NSP* に対応しています（リモート開発環境では、これは UNIX および OpenVMS プラットフォームにのみ適用され、オブジェクトが Adabas に保存されるメインフレームプラットフォームには適用されません）。

ローカルプロジェクトフォルダに追加された Natural オブジェクトから行番号が削除されています。非アクティブ環境のワークスペースを見ると、ライブラリフォルダに緑の記号が表示されていることを確認できます。これは行番号省略の記号です。



この時点でリポジトリをチェックすると（CVS と Subversion の両方に対応したブラウザで ViewCVS を起動するなど）、追加したオブジェクトは *SRC* フォルダにはまだ表示されず、このオブジェクトのバージョンはまだ作成されていません。追加したオブジェクトをソース制御システムで見えるようにするには、オブジェクトをコミットする必要があります。これについては、このチュートリアルで後述します。

- 10 ライブラリ *LV-TEST* のノードを選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Status Library]** または **[Local Versioning]** > **[CVS Status Library]** を選択します。

コマンドプロンプトウィンドウが開き、オブジェクト *PGM1* がローカルに追加されたことが示されます。

Subversion の例：


```
A SRC\PGM1.NSP
```

CVS の例：

```
cvs status: Examining .  
cvs status: Examining SRC  
File: PGM1.NSP          Status: Locally Added
```


- 11 Enter キーを押して次に進みます。
- 12 ライブラリ LV-TEST で別の Natural オブジェクトを選択し（このチュートリアルでは、SUBPGM1 という名前のサブプログラムを追加すると仮定します）、コンテキストメニューの コマンド **[Local Versioning]** > **[SVN Add Object]** または **[Local Versioning]** > **[CVS Add Object]** を選択して、別の Natural モジュールをプロジェクトフォルダにコピーすることでソース制御システムに追加します。

プログラムエディタを使用して PGM1 に小さな変更を加え、この変更を保存します。次に PGM1 を選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[Copy to Project Folder]** を選択します。プログラムがシステムファイルからプロジェクトフォルダにコピーされます。プログラムがソース制御システムにすでに追加されているので、このコマンドを使用できます。

 **Note:** 別の方法として、コマンド **[Local Versioning]** > **[SVN Add Object]** または **[Local Versioning]** > **[CVS Add Object]** を選択することもできます。この場合は、ソース制御システムへの追加は無視されますが、オブジェクトはプロジェクトフォルダにコピーされます。

追加したすべてのオブジェクトを 1 ステップでソース制御システムにコミットします。

- 13 ライブラリ LV-TEST のノードを選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Commit Library]** または **[Local Versioning]** > **[CVS Commit Library]** を選択します。

 **Note:** コマンド **[SVN Commit Object]** または **[CVS Commit Object]** を使用して、各オブジェクトを個別にコミットすることもできます。または、1 ステップでオブジェクトを追加およびコミットする場合は、コマンド **[SVN Add and Commit Object]** または **[CVS Add and Commit Object]** を使用します。この場合は、単一のオブジェクトそれぞれに対して異なるコミットログメッセージを入力できます。

ログメッセージを入力するダイアログボックスが表示されます。現在の日時を含む文字列およびユーザー ID が自動的に入力されます。

- 14 次に進むには **[OK]** ボタンを選択します。

**[Show command scripts]** グループボックスのオプションが有効になっているため、コマンドプロンプトウィンドウが表示され、ソース制御システムによって実行されたコマンドおよびその実行結果が示されます。

Subversion の例：

```
C:\natural\natapps\SVN\LV-MyFolder>svn update LV-TEST
At revision 651.
C:\natural\natapps\SVN\LV-MyFolder>svn commit -m "Commit_date:2007-06-25_time:15:31:56_uid:USER1" LV-TEST
Adding LV-TEST\SRC\PGM1.NSP
Adding LV-TEST\SRC\SUBPGM1.NSN
Transmitting file data ..
Committed revision 652.
```

CVS の例：

```
cvs update: Updating LV-TEST
cvs update: Updating LV-TEST/SRC
A LV-TEST/SRC/PGM1.NSP
A LV-TEST/SRC/SUBPGM1.NSN

cvs commit: Examining LV-TEST
cvs commit: Examining LV-TEST/SRC
/qatest1/Fuser200/LV-TEST/SRC/PGM1.NSP,v <-- LV-TEST/SRC/PGM1.NSP
initial revision: 1.1
/qatest1/Fuser200/LV-TEST/SRC/SUBPGM1.NSN,v <-- LV-TEST/SRC/SUBPGM1.NSN
initial revision: 1.1
```

LV-TEST ライブラリツリーをリポジトリと同期するため、Subversion または CVS の commit コマンドの前に update コマンドが発行されます。

CVS では、PGM1.NSP と SUBPGM1.NSN がそれぞれバージョン 1.1 でコミットされていることを確認できます。Subversion では、PGM1.NSP と SUBPGM1.NSN が新しいリビジョンでコミットされていることを確認できます。上記の例ではリビジョン 652 です。この時点でオブジェクトはソース制御システム内に配置されています。

この時点でリポジトリをチェックすると（CVS と Subversion の両方に対応したブラウザで ViewCVS を起動するなど）、これらのオブジェクトがフォルダ LV-MyModule/LV-TEST/SRC に含まれていることを確認できます。

### ソース制御システムのプロジェクトのブランチ作成

この時点で、プロジェクトの最初のバージョンは準備が整い、モジュール *LV-MyModule* にコミットされています。このモジュールからブランチを作成し、"Beta1" と呼ぶことにします。このブランチは、どのリビジョンのどのオブジェクトがプロジェクトの最初のバージョンに含まれているかを常に示します。今後プロジェクトの最初のバージョンに変更を加える必要ができた場合は（このバージョンのバグが報告された場合など）、Beta1 ブランチを使用します。当面の間は、メインブランチでプロジェクトの第2バージョンの開発を続けることができます。

CVSでは、ブランチはCVSコマンド `tag` で作成します。Subversionでは、ブランチはSubversionコマンド `copy` で作成します。ブランチを作成する前に、このブランチの名前 (Beta1) をプロジェクト定義に入力する必要があります。

#### ▶手順 3.6. ブランチを作成するには

- 1 前の演習で使用した**プロジェクト定義**のダイアログボックスを開き、**[Branch]** テキストボックスに名前「Beta1」を入力します。他のすべてのエントリを変更せずにおきます。
- 2 Naturalライブラリワークスペースで、システムファイルのノードを選択し、コンテキストメニューを表示し、カスケードメニュー **[Local Versioning]** のコマンド **[SVN Make Branch]** または **[CVS Make Branch/Tag]** を選択します。



**Note:** プロジェクト定義でブランチが指定されているため、これらのコマンドは有効になっています。

**[Show command scripts]** グループボックスのオプションが有効になっているため、モジュール *LV-MyModule* のすべてのオブジェクトに "Beta1" タグが付いていることを確認できます。

Subversion の例：

```
C:\natural\natapps\SVN\LV-MyFolder>svn copy http://natsvn/repos/user1/LV-MyModule
http://natsvn/repos/user1/Beta1/ -m "make branch"

Committed revision 653.
Press any key to continue . . .
```

CVS の例：

```
$ cvs -d :pserver:natsvn:/qatest1 tag -b Beta1
cvs tag: Tagging
cvs tag: Tagging LV-TEST
cvs tag: Tagging LV-TEST/SRC
T LV-TEST/SRC/PGM1.NSP
T LV-TEST/SRC/SUBPGM1.NSN
```



**Note:** 次の演習では、Beta1 ブランチではなく、メインブランチで作業を続行します。Beta1 ブランチで作業するには、Beta1 ブランチをチェックアウトしておく必要があります。また、メインブランチと他のブランチを混同しないように、Beta1 ブランチ用に別のプロジェクトフォルダを指定する必要があります。これについては、このチュートリアルで後述します。

## メインブランチでの作業

ブランチ Beta1 の作成が完了しました。ここからはメインブランチでプロジェクトの作業を続行します。これを行うには、プロジェクト定義からブランチ Beta1 を削除する必要があります。

メインバージョンのプロジェクトフォルダの削除や変更は行っていないため、メインバージョンをチェックアウトする必要はありません。オブジェクトはすでにそこにあります。プロジェクト定義に入力したプロジェクトフォルダがローカルプロジェクトフォルダと一致していることを確認します。また、Natural 環境の Natural オブジェクトとプロジェクトフォルダの Natural オブジェクトが同一のままであることを確認する必要があります。チェックアウト後にオブジェクトが同一でなかった場合は、ローカルバージョン管理の **[Import Library into Natural]** コマンドを使用してモジュールをインポートする必要があります。

### ▶手順 3.7. メインブランチで作業を続行するには

- 1 前の演習で使用した**プロジェクト定義**のダイアログボックスを開き、**[Branch]** テキストボックスから名前 "Beta1" を削除します。他のすべてのエントリを変更せずにおきます。
- 2 FUSER にプロジェクトの最新のバージョンがあることを確認するため、ライブラリ LV-TEST を選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Check Out and Import Library]** または **[Local Versioning]** > **[CVS Check Out and Import Library]** を選択します。

既存の Natural ソースごとに、オブジェクトを置き換えるかどうかを確認するメッセージが表示されます。

- 3 オブジェクトを置き換えるには、両方のオブジェクトで **[Yes]** ボタンを選択します。
- 4 プロジェクトの開発を続けるには、プログラム PGM1 に小さな変更を加えて保存または格納します。
- 5 この変更をソース制御システムに適用するには、オブジェクト PGM1 を選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[Copy to Project Folder]** を選択します。

- 6 プロジェクトフォルダのステータスをチェックするには、ライブラリ LV-TEST を選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Status Library]** または **[Local Versioning]** > **[CVS Status Library]** を選択します。

Subversion を使用している場合は、SRC\PGM1.NSP に "M" (変更済み) のマークが付きます。SUBPGM1 は最新なのでリストに表示されません。

CVS を使用している場合は、PGM1 のステータスは "Locally Modified"、SUBPGM1 のステータスは "Up-to-date" です。

これは、ライブラリをソース制御システムに最後にコミットした後、PGM1 を変更しましたが、SUBPGM1 は変更していないので、正常なステータスです。

必要な変更はすべて完了したので、これをコミットできます。

- 7 ライブラリ LV-TEST のノードを選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Commit Library]** または **[Local Versioning]** > **[CVS Commit Library]** を選択します。

ログメッセージを入力するダイアログボックスが表示されます。

- 8 次に進むには **[OK]** ボタンを選択します。

Subversion では、新しいリビジョンが作成されます。この例ではリビジョン 654 です。このリビジョンには、変更した PGM1 と古い SUBPGM1 が含まれています。これはプロジェクトの新しいバージョンです。リビジョン 653 が含まれる Beta1 ブランチとは異なります。

CVS では、PGM1 の新しいバージョンであるバージョン 1.2 が作成されます。これで、"Beta1" のタグの付いたバージョンとは別に、プロジェクトの新しいバージョンが作成されました。実際に異なっているのは PGM1 だけです。SUBPGM1 はブランチ Beta1 とメインブランチに属しています。

### 別のブランチでの作業

バージョン Beta1 を顧客に出荷し、顧客がバグを報告してきたため、それを修正したいとします。顧客がブランチ Beta1 を使用していることはわかっています。このブランチのソースが必要なため、プロジェクト定義を再度変更する必要があります。この場合は、ブランチ名に加えて、新しいプロジェクトフォルダを指定する必要があります。メインブランチと Beta1 ブランチを混同しないように、新しいプロジェクトフォルダを指定する必要があります。

#### ▶手順 3.8. ブランチ Beta1 で作業するには

- 1 前の演習で使用した **プロジェクト定義** のダイアログボックスを開き、次の手順に従います。
  - Subversion では、**[Module]** テキストボックスのリポジトリモジュール "LV-MyModule" を名前「Beta1」で置き換えます。

CVS では、**[Branch]** テキストボックスに名前「Beta1」を入力します。

- Subversion と CVS の両方で、**[Project folder]** テキストボックスに名前「LV-MyFolder-Beta1」を入力します。
  - 2 コンフィグレーションユーティリティを使用して、ワークスペースを追加の非アクティブ環境として定義します（例えば、Subversion のワークスペース内のプロジェクトフォルダ `C:\natural\natapps\SVN\LV-MyFolder-Beta1` をポイントする DBID 30、FNR 32 など）。
- 新しい非アクティブ環境が Natural スタジオのライブラリワークスペースに表示されます。
- 3 ライブラリ LV-TEST に古い Natural オブジェクトがないことを確認するため、このライブラリを FUSER から削除します。
  - 4 システムファイルのノードを選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Check Out and Import System File]** または **[Local Versioning]** > **[CVS Check Out and Import System File]** を選択します。

Subversion：リポジトリモジュール *Beta1*（モジュール *LV-MyModule* と並行して同じ内容で作成）内のオブジェクトがチェックアウトされます。リビジョン 653 の PGM1 と SUBPGM1 がプロジェクトフォルダ *LV-MyFolder-Beta1* に読み込まれ、FUSER 上のライブラリ LV-TEST にインポートされます。この例では、プロジェクトのメインブランチにある PGM1 と SUBPGM1 の最新バージョンはリビジョン 654 である点に注意してください。

CVS：Beta1 ブランチのオブジェクトがチェックアウトされます。PGM1（リビジョン 1.1）と SUBPGM1（リビジョン 1.1）がプロジェクトフォルダ *LV-MyFolder-Beta1* に読み込まれ、FUSER 上のライブラリ LV-TEST にインポートされます。プロジェクトのメインブランチにある PGM1 の最新バージョンはリビジョン 1.2 である点に注意してください。

Natural 環境で、ライブラリ LV-TEST が FUSER 上とワークスペースをポイントする非アクティブ環境に作成されます。SUBPGM1 は、メインブランチと Beta1 ブランチで同一です。PGM1 は、Beta1 後にソースの変更があったため異なります。

- 5 変更を PGM1 に適用します（バグを修正します）。
- 6 変更をソース制御システムに適用するには、オブジェクト PGM1 を選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[Copy to Project Folder]** を選択します。

変更したプログラム PGM1 は、フォルダ *LV-MyFolder-Beta1* にコピーされます。このフォルダは、前に Beta1 ブランチに指定したフォルダです。

- 7 プロジェクトフォルダのステータスをチェックするには、ライブラリ LV-TEST を選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Status Library]** または **[Local Versioning]** > **[CVS Status Library]** を選択します。

Subversion を使用している場合は、PGM1 のステータスは "M"（変更済み）です。SUBPGM1 は変更されていないのでリストに表示されません。

CVS を使用している場合は、PGM1 のステータスは "Locally Modified" です。SUBPGM1 のステータスは "Up-to-date" です。

Beta1 ブランチに必要な変更はすべて完了したので、これをコミットします。



- 8 ライブラリ LV-TEST のノードを選択し、コンテキストメニューを表示し、コマンド **[Local Versioning]** > **[SVN Commit Library]** または **[Local Versioning]** > **[CVS Commit Library]** を選択します。

ログメッセージを入力するダイアログボックスが表示されます。

- 9 次に進むには **[OK]** ボタンを選択します。

**[Library commands]** オプション ( **[Show command scripts]** グループボックス) が有効になっているため、リポジトリに何が起こるかを確認できます。

Subversion では、PGM1 の新しいバージョンと SUBPGM1 のバージョンを含む新しいリビジョン (この例では 655) が作成されます。ViewCVS で Subversion 情報を表示すると、リポジトリの *Beta1* モジュールに Beta1 バージョンがあることを確認できます。

CVS では、リビジョン 1.1.2.1 の PGM1 の新しいバージョンが作成されます。このプログラムはメインブランチではリビジョン 1.2 である点に注意してください。ViewCVS で CVS 情報を表示すると、CVS でリビジョン 1.1 からブランチが開始されていることを確認できます。


この時点でブランチ Beta1 の最新のバージョンがソース制御システムに保存されています。これでチュートリアルは終了です。

## ローカルバージョン管理の操作

---

このセクションでは、ローカルバージョン管理で使用できるすべてのコマンドについて説明します。次のトピックがあります。

- **新規プロジェクトの設定**
- **ソース制御システムへのライブラリの追加**
- **ソース制御システムへのオブジェクトの追加**
- **プロジェクトフォルダへのオブジェクトのコピーとコミット**
- **ソース制御システムからの既存オブジェクトの取得**
- **Natural へのインポート**
- **プロジェクトフォルダとリポジトリの同期**
- **ソース制御システムからのオブジェクトの削除**
- **ライブラリまたはシステムファイルのステータスの表示**
- **ログの表示**
- **ブランチの作成**

 **Note:** ローカルバージョン管理に実装されているコマンドを発行すると、選択したすべての *Natural* オブジェクトが処理されます。複数のオブジェクトを選択した場合 (プログラムノードのリストビューウィンドウなどで) は、ローカルバージョン管理コマンドは選択したすべてのオブジェクトに適用されます。 **[Show command scripts]** グループボッ




クスのオプションが有効になっている場合は、選択したオブジェクトごとにコマンドプロンプトウィンドウが表示されます。

## 新規プロジェクトの設定

新規モジュールに新規プロジェクトを設定するには、コマンド **[SVN Import Module into Repository]** または **[CVS Import Module into Repository]** を使用します。このコマンドは次の情報を使用します。


- プロジェクト定義で指定されたリポジトリがソース制御システムでアクセスされます。
- プロジェクト定義で指定された名前のモジュールがリポジトリに作成されます。
- プロジェクト定義で指定されたとおりにワークスペースフォルダとプロジェクトサブフォルダが作成されます。

新規プロジェクトを開始するには、プロジェクトフォルダが空である必要があります。プロジェクトフォルダがすでに存在し、そこにモジュールが含まれている場合は、これらのモジュールはソース制御システムに追加されます（これは、Subversion または CVS のドキュメントでは「ベンダーブランチ」と呼ばれます）。

 **Note:** ソース制御システムから既存のプロジェクトを取得する場合は、コマンド **[SVN Check Out System File]** または **[CVS Check Out System File]** と、コマンド **[Import System File into Natural]** を使用します。この2つの手順を1つのコマンド **[SVN Check Out and Import System File]** または **[CVS Check Out and Import System File]** で実行することもできます。詳細については、「[ソース制御システムからの既存オブジェクトの取得](#)」および「[Natural へのインポート](#)」を参照してください。

### ▶手順 3.9. 新規プロジェクトを設定するには

- 1 ソース制御システムに保存するソースを含むシステムファイルのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Import Module into Repository]** または **[Local Versioning]** > **[CVS Import Module into Repository]** を選択します。

 **Note:** このコマンドの使用は、**オプション**で無効にできます。

### ソース制御システムへのライブラリの追加

ライブラリを CVS に追加すると、直ちに CVS 内で参照できるようになります。オブジェクトとは異なり、ライブラリをコミットする必要はありません。Natural ライブラリの CVS ディレクトリは、コミットしなくても CVS に作成されます。


CVS とは異なり、また CVS および Subversion のオブジェクトと同様に、Subversion ではライブラリを追加後にコミットしないと参照できるようになりません。

#### ▶手順 3.10. ライブラリを追加するには

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Add Library]** または **[Local Versioning]** > **[CVS Add Library]** を選択します。
- 3 Subversion のみ：コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Commit Library]** を選択します。

#### ▶手順 3.11. ライブラリを追加してコミットするには (Subversion のみ)


- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Add and Commit Library]** を選択します。

 **Note:** ライブラリを追加するコマンドは、**オプション**で無効にできます。

### ソース制御システムへのオブジェクトの追加

オブジェクトをソース制御システムに追加するには、まずオブジェクトを追加し、次にそのオブジェクトをコミットする必要があります。また、オブジェクトの追加とコミットを 1 ステップで実行することも、ライブラリ全体の内容をコミットすることも、複数のライブラリを持つシステムファイルをコミットすることもできます。

オブジェクトをソース制御システムに追加すると、そのオブジェクトはプロジェクトフォルダにコピーされます。

 **Tip:** ライブラリワークスペースでファイルビューに切り替え、ライブラリの **[Src]** (ソース) ノードを開いてリストビューウィンドウを表示すると、このライブラリのすべてのオブジェクトを 1 ステップで追加できます。また、複数のオブジェクトを 1 つのコマンド **[SVN Commit Object]** または **[CVS Commit Object]** で一度にコミットすることもできます。これらのコマンドでは、オブジェクトごとに異なるコミットコメントをコミットログメッセージに記録できます (選択したオブジェクトごとにダイアログボックスが表示されます)。オブジェクトごとに異なるコミットコメントを記録する必要がない場合は、1 つのコマンド **[SVN Commit Library]** または **[CVS Commit Library]** を使用して複数のオブジェクトを一度にコミットします。

**▶手順 3.12. オブジェクトを追加するには**

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Add Object]** または **[Local Versioning] > [CVS Add Object]** を選択します。

**▶手順 3.13. オブジェクトをコミットするには**

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Commit Object]** または **[Local Versioning] > [CVS Commit Object]** を選択します。

ログメッセージを入力するダイアログボックスが表示されます。存在する場合は、オブジェクトが最後に保存された日時を含む文字列と、オブジェクトを保存したユーザーの ID が自動的に指定されます。セーブデータはリソースおよびエラーメッセージでは使用できません。

- 3 **[OK]** ボタンを選択します。

**▶手順 3.14. オブジェクトの追加とコミットを 1 ステップで実行するには**

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Add and Commit Object]** または **[Local Versioning] > [CVS Add and Commit Object]** を選択します。

ログメッセージを入力するダイアログボックスが表示されます。存在する場合は、オブジェクトが最後に保存された日時を含む文字列と、オブジェクトを保存したユーザーの ID が自動的に指定されます。セーブデータはリソースおよびエラーメッセージでは使用できません。

- 3 **[OK]** ボタンを選択します。

**▶手順 3.15. ライブラリ全体の内容をコミットするには**

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Commit Library]** または **[Local Versioning] > [CVS Commit Library]** を選択します。




**Note:** 上記のコマンドは、Subversion または CVS の update コマンドを実行してからライブラリをコミットします。

ログメッセージを入力するダイアログボックスが表示されます。現在の日時を含む文字列およびユーザー ID が自動的に入力されます。

- 3 **[OK]** ボタンを選択します。

### ▶手順 3.16. システムファイル全体の内容をコミットするには

- 1 システムファイルのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Commit System File]** または **[Local Versioning]** > **[CVS Commit System File]** を選択します。


 **Note:** 上記のコマンドは、Subversion または CVS の update コマンドを実行してからシステムファイルをコミットします。

ログメッセージを入力するダイアログボックスが表示されます。現在の日時を含む文字列およびユーザー ID が自動的に入力されます。

- 3 **[OK]** ボタンを選択します。


### プロジェクトフォルダへのオブジェクトのコピーとコミット

オブジェクトがすでにプロジェクトフォルダに 1 回コピーされている場合は（コマンド **[SVN Add Object]** または **[CVS Add Object]** を使用）、それ以降にオブジェクトに適用した変更は、コマンド **[Copy to Project Folder]** を使用してプロジェクトフォルダにコピーできます。

 **Note:** コマンド **[SVN Add Object]** または **[CVS Add Object]** を使用することもできます。

### ▶手順 3.17. オブジェクトをプロジェクトフォルダにコピーするには

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[Copy to Project Folder]** を選択します。

 **Note:** オブジェクトをまだプロジェクトフォルダに追加していない場合は、オブジェクトを追加するかどうかを確認するメッセージが表示されます。

### ▶手順 3.18. オブジェクトをプロジェクトフォルダにコピーしてコミットするには

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[Copy and SVN Commit Object]** または **[Local Versioning]** > **[Copy and CVS Commit Object]** を選択します。

## ソース制御システムからの既存オブジェクトの取得

ソース制御システムから既存のプロジェクトを取得する場合は、システムファイル全体をチェックアウトできます。ライブラリ1つだけをチェックアウトすることも可能です。CVSを使用している場合は、単一のオブジェクトをチェックアウトできます。Subversion では、単一のオブジェクトのチェックアウトはサポートされていません。

ライブラリをチェックアウトするときに、対応するプロジェクトフォルダがワークスペース内にまだ存在しない場合は、プロジェクトフォルダが作成されます。ただし、この場合はプロジェクトフォルダがバージョン制御下にないため、プロジェクトフォルダがバージョン制御下にあることを要求する一部のコマンド（**[SVN Commit System File]** または **[CVS Commit System File]**、**[SVN Display Log]** または **[CVS Display Log]** など）は実行できません。CVSを使用している場合は、オブジェクトをチェックアウトするときに、ライブラリが存在しないかバージョン制御下にないと、同じ制限が適用されます。

オブジェクトをチェックアウトすると、ファイルがソース制御システムからローカルワークスペースのプロジェクトフォルダにコピーされます。

### ▶手順 3.19. システムファイルをチェックアウトするには

- 1 システムファイルのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Check Out System File]** または **[Local Versioning]** > **[CVS Check Out System File]** を選択します。

### ▶手順 3.20. ライブラリをチェックアウトするには

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Check Out Library]** または **[Local Versioning]** > **[CVS Check Out Library]** を選択します。

### ▶手順 3.21. オブジェクトをチェックアウトするには (CVS のみ)

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[CVS Check Out Object]** を選択します。

### Natural へのインポート

オブジェクトをリポジトリからチェックアウトし、Natural 内（通常は FUSER 内）でそのオブジェクトに対して作業を行うには、オブジェクトを Natural にインポートする必要があります。チェックアウトとインポートを組み合わせたコマンドを使用することも、ワークスペース内のプロジェクトフォルダにオブジェクトをチェックアウトしてからいずれかのインポートコマンドを使用することも可能です。

#### ▶手順 3.22. システムファイルのチェックアウトとインポートを 1 ステップで実行するには

- 1 システムファイルのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Check Out and Import System File]** または **[Local Versioning] > [CVS Check Out and Import System File]** を選択します。

#### ▶手順 3.23. システムファイルをインポートするには

- 1 システムファイルのノードを選択します。このシステムファイルはチェックアウトされている必要があります。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Import System File into Natural]** または **[Local Versioning] > [CVS Import System File into Natural]** を選択します。

#### ▶手順 3.24. ライブラリのチェックアウトとインポートを 1 ステップで実行するには

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Check Out and Import Library]** または **[Local Versioning] > [CVS Check Out and Import Library]** を選択します。

#### ▶手順 3.25. ライブラリをインポートするには

- 1 ライブラリのノードを選択します。このライブラリはチェックアウトされている必要があります。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Import Library into Natural]** または **[Local Versioning] > [CVS Import Library into Natural]** を選択します。

#### ▶手順 3.26. オブジェクトのチェックアウトとインポートを 1 ステップで実行するには（CVS のみ）

- 1 オブジェクトのノードを選択します。

- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[CVS Check Out and Import Object]** を選択します。

▶**手順 3.27. オブジェクトをインポートするには**

- 1 オブジェクトのノードを選択します。このオブジェクトはチェックアウトされている必要があります。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Import Object into Natural]** または **[Local Versioning]** > **[CVS Import Object into Natural]** を選択します。

## プロジェクトフォルダとリポジトリの同期

プロジェクトの作業をチームで行う場合は、作業中のソースを他の開発者が変更することがあります。更新コマンドを使用すると、リポジトリの変更内容が作業中のコピーに取り込まれます。

▶**手順 3.28. システムファイルを更新するには**

- 1 システムファイルのノードを選択します。このシステムファイルはバージョン制御下にある必要があります。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Update System File]** または **[Local Versioning]** > **[CVS Update System File]** を選択します。

▶**手順 3.29. ライブラリを更新するには**

- 1 ライブラリのノードを選択します。このライブラリはバージョン制御下にある必要があります。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Update Library]** または **[Local Versioning]** > **[CVS Update Library]** を選択します。


▶**手順 3.30. オブジェクトを更新するには**

- 1 オブジェクトのノードを選択します。このオブジェクトはバージョン制御下にある必要があります。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Update Object]** または **[Local Versioning]** > **[CVS Update Object]** を選択します。



### ソース制御システムからのオブジェクトの削除

ソース制御システムからオブジェクトを削除するには、まずオブジェクトを削除し、次にそのオブジェクトをコミットする必要があります。また、オブジェクトの削除とコミットを1ステップで実行することも、ライブラリ全体の内容を削除することもできます（ライブラリ自体を含めるか含めないかを選択できます）。

 **Caution:** 削除は強制削除として実行されます。変更済みのオブジェクトおよびバージョン制御下でないオブジェクトも削除されます。

必ず削除の確認を求められます。オブジェクトとライブラリはワークスペースから削除されるだけで、Natural 開発環境からは削除されません。

ただし、Natural 開発環境（通常はFUSER）とワークスペースが同一の場合は、オブジェクトはNatural 開発環境からも削除されます。この場合は、オブジェクトまたはライブラリは削除後に消失するため、オブジェクトまたはライブラリの削除とコミットを1つのコマンドで実行できます。または、1つ上の階層のノードを使用して削除をコミットします（つまり、システムファイルノードを使用してライブラリをコミットし、ライブラリノードを使用してオブジェクトを削除します）。

#### ▶手順 3.31. オブジェクトを削除するには

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、[Local Versioning] > [SVN Delete Object] または [Local Versioning] > [CVS Remove Object] を選択します。
- 3 削除したオブジェクトをコミットするには、コンテキストメニューを表示し、[Local Versioning] > [SVN Commit Object] または [Local Versioning] > [CVS Commit Object] を選択します。

#### ▶手順 3.32. オブジェクトの削除とコミットを1ステップで実行するには

- 1 オブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、[Local Versioning] > [SVN Delete and Commit Object] または [Local Versioning] > [CVS Remove and Commit Object] を選択します。

#### ▶手順 3.33. ライブラリを削除するには

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、[Local Versioning] > [SVN Delete Library] または [Local Versioning] > [CVS Remove Library] を選択します。
- 3 削除したライブラリをコミットするには、コンテキストメニューを表示し、[Local Versioning] > [SVN Commit Library] または [Local Versioning] > [CVS Commit Library] を選択します。



**▶手順 3.34. ライブラリの削除とコミットを1ステップで実行するには**

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Delete and Commit Library]** または **[Local Versioning] > [CVS Remove and Commit Library]** を選択します。

**ライブラリまたはシステムファイルのステータスの表示**

プロジェクトフォルダのライブラリまたはシステムファイルのステータスを表示すると、オブジェクトが変更されているかどうかを確認できます。

Subversion では、変更されたオブジェクトは "M" でマークされます。CVS では、変更されたオブジェクトのステータスは "Locally Modified" になります。最後に行った変更は、ソース制御システムにまだコミットされていません。

オブジェクトのステータスがリストに表示されない場合 (Subversion)、またはステータスが "Up-to-date" の場合 (CVS) は、オブジェクトは最後にコミットされてから変更されていません。



**Note:** その他のステータスの説明については、Subversion または CVS のドキュメントを参照してください。

**▶手順 3.35. ライブラリのステータスを確認するには**

- 1 ライブラリのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Status Library]** または **[Local Versioning] > [CVS Status Library]** を選択します。

**▶手順 3.36. システムファイルのステータスを確認するには**

- 1 システムファイルのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning] > [SVN Status System File]** または **[Local Versioning] > [CVS Status System File]** を選択します。

### ログの表示

ファイルの履歴情報、つまりコミットログメッセージを出力できます。

#### ▶手順 3.37. ログを表示するには


- 1 システムファイル、ライブラリ、またはオブジェクトのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Display Log]** または **[Local Versioning]** > **[CVS Display Log]** を選択します。

### ブランチの作成

ブランチを作成できるのは、ブランチ名がプロジェクト定義に指定されている場合に限られます。


Subversion：ブランチを作成すると、指定したモジュール内のすべてのオブジェクトが別のモジュールにコピーされます（モジュール名がブランチ名の場合）。

CVS：ブランチを作成すると、指定したモジュール内のすべてのオブジェクトに、プロジェクト定義で指定したブランチ名のタグが付けられます。

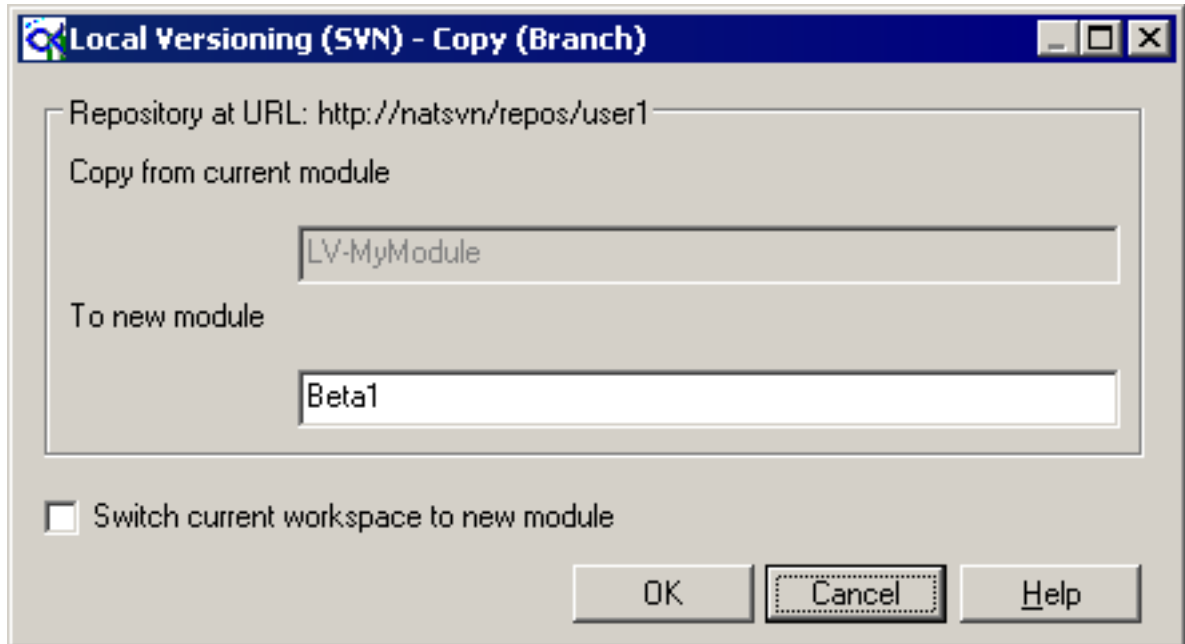
 **Note:** 新しいブランチで作業するには、チェックアウトしておく必要があります。ブランチごとに異なるプロジェクトフォルダを使用することをお勧めします。


#### ▶手順 3.38. ブランチを作成するには

- 1 システムファイルのノードを選択します。
- 2 コンテキストメニューを表示し、**[Local Versioning]** > **[SVN Make Branch]** または **[Local Versioning]** > **[CVS Make Branch/Tag]** を選択します。

 **Note:** このコマンドの使用は、オプションで無効にできます。

Subversion のみ：次のダイアログボックスが表示されます。



 **Note:** CVS の場合は、タグ付けのみが行われ、新規モジュールへのコピーは行われな  
いため、このダイアログボックスは表示されません。また、CVS では `switch` コマ  
ンドを使用できません。

- 3 ダイアログボックスで新規モジュール名（ブランチ）を指定します。

また、**[Switch current workspace to new module]** チェックボックスをオンにできます。  
リポジトリアドレスは `.svn` ステータスフォルダに保存されるため、1つのワークスペースは  
1つのリポジトリにしか使用できない点に注意してください。新規モジュールに切り替える  
と、ワークスペースは新しいリポジトリアドレスで更新されます。

- 4 **[OK]** ボタンを選択します。

## ログの処理

ローカルバージョン管理では、ログファイルが作成されます。このファイルは、システムライブ  
ラリ `SYSPLCVS` にリソース（テキストドキュメント）として保存されます。このファイルの名前  
は `lv_log.log` です。このログファイルには、処理されたステップの情報、および場合によっては  
エラー情報が記録されます。

複数のオブジェクトの処理中に何らかの理由でローカルバージョン管理が停止した場合は、この  
ログファイルを使用してください。ログファイルでは、最後に処理されたオブジェクトを確認  
できます。

## コマンドスクリプト

---

コンテキストメニューのローカルバージョン管理コマンド（**[SVN Commit Library]** や **[CVS Commit Library]** など）は、Windows のコマンドスクリプトで制御されます。これらのコマンドスクリプトは、システムライブラリ SYSPLCVS にリソースとして保存されています。

# 4 Natural for Ajax Tools

---

▪ 開始する前に .....	58
▪ Natural for Ajax Tools の要素 .....	59
▪ サーバーの設定 .....	60
▪ ライブラリとプロジェクトの設定 .....	61
▪ Application Designer 開発ワークスペースを開く .....	61
▪ 生成されたアダプタのインポート .....	62
▪ アダプタからのメインプログラムの生成 .....	63
▪ ブラウザでのメインプログラムの実行 .....	63
▪ Natural プログラムまたは Application Designer 開発ワークスペースの URL の表示 .....	64

Natural for Ajax がインストールされている場合は、Natural for Ajax Tools を使用して、Natural スタジオから Natural for Ajax の機能を直接使用できます。Natural for Ajax Tools を使用すると、豊富なインターネットアプリケーション用の Web ページを設計し、これらのページのアダプタを Natural ライブラリにインポートしてこれらのページを表示するプログラムを生成し、内部ブラウザでこれらのプログラムをテストできます。

このchapterでは、次のトピックについて説明します。

詳細については、『*Natural for Ajax*』ドキュメントを参照してください。

## 開始する前に

---

Natural for Ajax Tools は、Natural スタジオのオプションプラグインです。このプラグインのインストールは Natural のインストールの一部です。このプラグインは初期インストール時にインストールすることも、または後で別のセットアッププロセスとしてインストールすることもできます。インストール手順の詳細については、『インストール』ドキュメントを参照してください。

Natural for Ajax Tools を使用する前に、次のトピックの説明に従う必要があります。

- **前提条件**
- **Natural for Ajax Tools の有効化**

### 前提条件

プラグインを有効にすると、Natural スタジオが、パラメータファイル NATPARM を使用して別の Natural セッションを開始します。Natural セッションで NATPARM 以外のパラメータファイルを使用している場合は、FNAT（特にライブラリ SYSPLNIX および SYSLIBS）や他の Natural パラメータ（WORK パラメータなど）が Natural プラグインセッション（NATPARM を使用）と Natural セッション（別のパラメータファイルを使用）が一致しないことがある点に注意してください。このために予測できないエラーが発生する場合があります。このため、Natural セッションでパラメータファイル NATPARM も使用することをお勧めします。Natural パラメータファイルの詳細については、『*Natural Security*』ドキュメントの「*Natural Security* でのプラグイン」を参照してください。

## Natural for Ajax Tools の有効化

Natural for Ajax Tools をインストールしたら、プラグインマネージャで有効にする必要があります。有効にするための手順の詳細については、『*Natural スタジオの使用*』ドキュメントの「プラグインマネージャ」を参照してください。





## Natural for Ajax Tools の要素


プラグインマネージャで Natural for Ajax Tools を有効にすると、Natural スタジオウィンドウで次の追加要素が使用可能になります。

- ツールバー
- コンテキストメニュー

### ツールバー

追加のツールバーが表示されます。このツールバーのボタンは、次のメニューコマンドを表します。

-  **Import Adapters**
-  **Create Main Program**
-  **Show URL**
-  **Execute in Browser**
-  **Open Application Designer**

 **Note:** このツールバーは、[ツール] メニューの [カスタマイズ] を選択して表示と非表示を切り替えることができます。

### コンテキストメニュー

現在のコンテキスト（選択したオブジェクト）が対応している場合は、コンテキストメニューにカスケードメニュー [Natural for Ajax] が追加されます。カスケードメニューに含まれるコマンドは、コンテキストに応じて変化します。

#### ライブラリレベル

ライブラリを選択しているときは、カスケードメニューに次のコマンドが表示されます。

- **Import Adapters**
- **Open Application Designer**

**Natural** オブジェクトレベル

プログラムを選択しているときは、カスケードメニューに次のコマンドが表示されます。

- **Show URL**
- **Execute in Browser**

アダプタを選択しているときは、カスケードメニューに次のコマンドが表示されます。

- **Create Main Program**

## サーバーの設定

---

Natural for Ajax アプリケーションを開発するには、Natural 開発サーバー（NDV、オプション）、Natural for Ajax を実行するアプリケーションサーバー、および Web I/O インターフェイス（NWO）が必要です。

Natural for Ajax Tools をプラグインマネージャで有効にした後、最初に Natural 開発サーバーおよび対応するアプリケーションサーバーと Web I/O インターフェイスサーバーとの関係を設定する必要があります。設定は、ライブラリ SYSPLNJBX 内にリソースとして用意されているコンフィグレーションファイル *ConfigNJXPLG.dat* で行います。

コンフィグレーションデータの各エントリは、次の情報で構成されています。

- Natural 開発サーバーのホスト名
- Natural 開発サーバーのポート番号（省略可能）
- アプリケーションサーバーのホスト名
- アプリケーションサーバーのポート番号
- アプリケーションサーバーのパス名
- Web I/O インターフェイスサーバーのホスト名
- Web I/O インターフェイスサーバーのポート番号

詳細については、*ConfigNJXPLG.dat* ファイルの内容を参照してください。

このファイルには、ローカル環境で作業する場合に必要な設定がすでに含まれています。新規リモート開発環境のコンフィグレーション行を定義した後、対応するアプリケーションサーバー上のファイル *sessions.xml*（『オペレーション』ドキュメントの「コンフィグレーションファイルの内容」を参照）に同名のセッション定義が含まれていることを確認してください。

例：



`ConfigNJXPLG.dat` ファイルには、次の行が含まれています。

```
MYNDV,2700,MYAPPS,8080,\\MYAPPS\JB0SS\#LIB\nat,MYNW0,2900
```

この場合は、"MYAPPS:8080" で実行されているアプリケーションサーバーの `sessions.xml` ファイルに次のセッション定義が含まれている必要があります。

```
<sessions>
  <session id="MYNDV">
    <natural_program>script-name</natural_program>
    <user id="" pwd=""/>
  </session>
</sessions>
```

オプションでユーザー ID とパスワードを指定できます。これは [Execute in Browser] コマンドを使用するとき便利です。Natural スタジオで Natural for Ajax プログラムを開始するたびにユーザー ID とパスワードを入力する必要がなくなります。

## ライブラリとプロジェクトの設定

一般に、Natural for Ajax プロジェクトは、サーバー部分を実装する Natural オブジェクトを含む 1 つまたは複数の Natural ライブラリと、ユーザーインターフェイス部分を実装する 1 つまたは複数の Application Designer プロジェクトで構成されています。

現在のバージョンの Natural for Ajax Tools では、Natural for Ajax プロジェクトが 1 つの Natural ライブラリと 1 つの Application Designer プロジェクトで構成され、Application Designer プロジェクトの名前が Natural ライブラリの名前を小文字に変換したものと同一であることを前提にしています。

## Application Designer 開発ワークスペースを開く

Application Designer を使用すると、機能豊富なインターネットアプリケーションの Web ページをデザインできます。Application Designer 開発ワークスペースは、Natural スタジオから直接開くことができます。

### ▶手順 4.1. Application Designer 開発ワークスペースを開くには


- 1 Application Designer プロジェクトに属する Natural ライブラリを選択します。
- 2 コンテキストメニューを表示し、[Natural for Ajax] > [Open Application Designer] を選択します。

Or:

Natural for Ajax Tools のツールバーが表示されたら、次のツールバーボタンを選択します。



ブラウザウィンドウに Application Designer 開発ワークスペースが開きます。Web ページの Application Designer プロジェクトの作成、ページの作成、既存のページの編集、およびページの Natural アダプタの作成が可能になります。

 **Important:** Application Designer プロジェクトに、対応する Natural ライブラリの名前を小文字に変換した名前を付けます。

詳細については、『Natural for Ajax』ドキュメントの「*Developing the User Interface*」を参照してください。

## 生成されたアダプタのインポート

---

Natural アダプタは、Application Designer で Natural ページのレイアウトを保存したときに生成されます。生成されたアダプタは、アプリケーションサーバーから Natural ライブラリに同じ名前でインポートできます。

### ▶手順 4.2. 生成されたアダプタをインポートするには

- 1 生成されたアダプタをインポートするライブラリを選択します。
- 2 コンテキストメニューを表示し、[Natural for Ajax] > [Import Adapters] を選択します。

Or:

Natural for Ajax Tools のツールバーが表示されたら、次のツールバーボタンを選択します。



Application Designer プロジェクトと同じ名前の生成されたアダプタが、選択したライブラリにインポートされます。

ライブラリにアダプタがすでに存在する場合は、アダプタを置き換えるかどうかを確認するメッセージが表示されます。

## アダプタからのメインプログラムの生成

---

アダプタを呼び出してページを表示する Natural メインプログラムを、アダプタから生成できます。

### ▶手順 4.3. アダプタからメインプログラムを生成するには

- 1 アダプタを選択します。
- 2 コンテキストメニューを表示し、**[Natural for Ajax] > [Create Main Program]** を選択します。

Or:

Natural for Ajax Tools のツールバーが表示されたら、次のツールバーボタンを選択します。



プログラムが生成されます。ソースコードはプログラムエディタウィンドウに表示されます。

- 3 名前を選択し、プログラムを格納します。

ページイベントの処理方法などの詳細については、『*Natural for Ajax*』ドキュメントの「*Developing the Application Code*」を参照してください。

## ブラウザでのメインプログラムの実行

---

アダプタから生成された Natural メインプログラムを実行できます。Natural スタジオ内でブラウザウィンドウが開き、プログラムをテストできます。

### ▶手順 4.4. メインプログラムを実行するには

- 1 メインプログラムを選択します。
- 2 コンテキストメニューを表示し、**[Natural for Ajax] > [Execute in Browser]** を選択します。

Or:

Natural for Ajax Tools のツールバーが表示されたら、次のツールバーボタンを選択します。



ブラウザウィンドウでプログラムが実行されます。

## Natural プログラムまたは Application Designer 開発ワークスペースの URL の表示

---

Natural プログラムまたは Application Designer 開発ワークスペースを Natural スタジオ外部で実行する場合は、実行に必要な URL を表示できます。

### ▶手順 4.5. プログラムの URL を表示するには

- 1 プログラムを選択します。
- 2 コンテキストメニューを表示し、**[Natural for Ajax]** > **[Show URL]** を選択します。

Or:

Natural for Ajax Tools のツールバーが表示されたら、次のツールバーボタンを選択します。



ダイアログボックスが開き、プログラムの URL が示されます。

コピーアンドペーストを使用して、Application Designer でサポートされているブラウザ（Internet Explorer、Mozilla Firefox など）に、この URL を入力できます。これにより、プログラムをブラウザ内（Natural スタジオで使用される内部ブラウザ以外）で直接実行できます。

### ▶手順 4.6. Application Designer 開発ワークスペースの URL を表示するには

- 1 プログラム以外の任意のオブジェクトを選択するか、ライブラリまたはシステムファイルを選択します。

- 2 Natural for Ajax Tools ツールバーで次のツールバーボタンを選択します。



ダイアログボックスが開き、Application Designer 開発ワークスペースの URL が示されます。

コピーアンドペーストを使用して、Application Designer でサポートされているブラウザ（Internet Explorer、Mozilla Firefox など）に、この URL を入力できます。これにより、Application Designer 開発ワークスペースをブラウザ内（Natural スタジオで使用される内部ブラウザ以外）で直接実行できます。



# 索引

---

## A

- Application Designer
  - Natural for Ajax Tools で URL を表示, 64
  - Natural for Ajax Tools で開く, 61

## C

- commit
  - Subversion/CVS にオブジェクトを, 46
- CVS
  - Natural スタジオで使用, 9
  - 現在のリポジトリタイプ, 17
  - CVS コマンド, 14

## N

- Natural for Ajax Tools
  - Natural スタジオの拡張機能, 57
- Natural Security
  - ローカルバージョン管理の使用, 10
- Natural スタジオ
  - 拡張機能の使用, 1

## O

- Object Compare
  - Natural スタジオの拡張機能, 3
  - オプション, 6

## S

- Subversion
  - Natural スタジオで使用, 9
- SVN
  - Natural スタジオで使用, 9
  - 現在のリポジトリタイプ, 17
  - SVN コマンド, 14
- SYSPLCVS
  - ローカルバージョン管理のシステムファイル, 11

## U

- URL
  - Natural for Ajax Tools で表示, 64

## あ

- アダプタ
  - Natural for Ajax Tools でインポート, 62
- アーカイブ
  - データエリア, 19

## い

- インポート
  - Natural for Ajax Tools でアダプタを, 62
  - Subversion/CVS モジュール, 45
  - 無効化された Subversion/CVS のコマンド, 20

## お

- オブジェクト
  - Natural for Ajax Tools のコマンド, 60
  - Subversion/CVS からチェックアウト, 49
  - Subversion/CVS に追加, 46
  - Subversion/CVS のコマンド, 16
  - 比較, 7
- オプション
  - Object Compare, 6
  - ローカルバージョン管理, 16

## か

- 開始
  - Object Compare, 7
- 拡張
  - Natural スタジオ, 1
  - 外部比較ツール, 7

## き

- 行番号
  - ローカルバージョン管理, 12

## く

- クラス
  - ソース制御システムを使用, 9
  - 比較, 3
- グローバルデータエリア
  - ソース制御システムを使用, 9
  - 比較, 3

## こ

## 構成

Natural for Ajax Tools, 60, 61

## コピー

オブジェクトを Subversion/ CVS プロジェクトフォルダに,  
48

## コピーコード

ソース制御システムを使用, 9  
比較, 3

## コマンドスクリプト

Subversion/ CVS コマンドの表示, 18  
ローカルバージョン管理, 56

## コンテキストメニュー

Natural for Ajax Tools, 59  
Object Compare, 5  
ローカルバージョン管理, 14

## さ

## 削除

ローカルバージョン管理のプロジェクト, 27

## サブプログラム

ソース制御システムを使用, 9  
比較, 3

## サブルーチン

ソース制御システムを使用, 9  
比較, 3

## サーバー

Natural for Ajax Tools による構成, 60

## し

## システムファイル

Subversion/ CVS からチェックアウト, 49  
Subversion/ CVS のコマンド, 15  
Subversion/ CVS のステータス, 53

## 新規

ローカルバージョン管理のプロジェクト, 22

## 実行

Natural for Ajax Tools によるプログラム, 63

## せ

## 生成

Natural for Ajax Tools によるプログラム, 63

## 選択

比較するオブジェクト, 7

## 前提条件

Natural for Ajax Tools, 58  
Object Compare, 4  
ローカルバージョン管理, 12

## そ

## ソース制御システム

Natural スタジオで使用, 9

## ソースヘッダー

ワークスペース内, 18

## た

## ダイアログ

ソース制御システムを使用, 9  
比較, 3

## ち

## チェックアウト

Subversion/ CVS から, 49

## チュートリアル

ローカルバージョン管理, 27

## つ

## 追加

Subversion/ CVS にオブジェクトを, 46  
ライブラリを Subversion/ CVS に, 46  
ローカルバージョン管理のプロジェクト, 22

## ツールバー

Natural for Ajax Tools, 59  
Object Compare, 5  
ローカルバージョン管理, 14

## て

## 定義

ローカルバージョン管理のプロジェクト, 20

## テキスト

ソース制御システムを使用, 9  
比較, 3

## データエリア

アーカイブ, 19

## と

## 特殊文字

ローカルバージョン管理, 13

## は

## バージョン管理

Natural オブジェクト, 9

## パラメータデータエリア

ソース制御システムを使用, 9  
比較, 3

## ひ

## 比較

Natural オブジェクト, 3, 7

## 表示

ローカルバージョン管理のプロジェクト, 20

## ふ

## ファンクション

ソース制御システムを使用, 9  
比較, 3

## ブラウザ

Natural for Ajax Tools, 63

## ブランチ

Subversion/ CVS に作成, 54  
Subversion/ CVS のプロジェクト用に作成, 40  
別のブランチでの作業, 42  
無効化された Subversion/ CVS のコマンド, 20



- メインプランチでの作業, 41
- ローカルバージョン管理, 27
- プログラム
  - Natural for Ajax Tools で URL を表示, 64
  - Natural for Ajax Tools で実行, 63
  - Natural for Ajax Tools で生成, 63
  - ソース制御システムを使用, 9
  - 比較, 3
- プロジェクト
  - Natural for Ajax Tools による構成, 61
  - Subversion/ CVS モジュールで設定, 33, 45
  - ローカルバージョン管理, 20
- プロジェクトフォルダ
  - ローカルバージョン管理, 25

## へ

- ヘルプルーチン
  - ソース制御システムを使用, 9
  - 比較, 3

## む

- 無視
  - ローカルバージョン管理でのファイル, 24

## め

- メニューバー
  - Object Compare, 5
  - ローカルバージョン管理, 14

## も

- モジュール
  - Subversion/ CVS インポート, 45
  - ローカルバージョン管理, 27

## よ

- 要素
  - Natural for Ajax Tools, 59
  - Object Compare, 5
  - ローカルバージョン管理, 13
- 呼び出し
  - Object Compare, 7

## ら

- ライブラリ
  - Natural for Ajax Tools による構成, 61
  - Natural for Ajax Tools のコマンド, 59
  - Subversion/ CVS からチェックアウト, 49
  - Subversion/ CVS に追加, 46
  - Subversion/ CVS のコマンド, 15
  - Subversion/ CVS のステータス, 53
  - Subversion/ CVS へのコミット, 46
  - 無効化された Subversion/ CVS のコマンド, 20

## ろ

- ログの処理
  - ローカルバージョン管理, 55

- ローカルデータエリア
  - ソース制御システムを使用, 9
  - 比較, 3
- ローカルバージョン管理
  - Natural スタジオの拡張機能, 9
  - オプション, 16
  - チュートリアル, 27

## わ

- ワークスペース
  - ローカルバージョン管理, 25

