



BCC32C コンパイラを 使ってみよう

2017年12月

エンバカデロ・テクノロジーズ

目次

BCC32C コンパイラを使う	. 1
BCC32C コンパイラのインストール	. 1
環境変数 C の設定	. 2
BCC32C の動作チェック	. 3
はじめての C++アプリケーションを コンパイルする	. 4
Hello World アプリケーション	. 4
その他のツール	. 6
C++Builder でもっと効率的に プログラミングする	. 7
C++Builder とは	. 7
C++Builder で Hello World アプリケーション作成を体験する	. 9
コンソールアプリケーションの新規作成	. 9
プロジェクトマネージャ1	11
エディタのコーディング支援機能1	12
コンパイルエラーを調べる1	14
デバッガを使う1	15

BCC32C コンパイラを使う

BCC32C コンパイラは、Windows 32-bit 向けの Clang ベースのコンパイラです。エンバカデロのビジュ アル C++開発環境 C++Builder に搭載されており、BCC64 コンパイラとともに、C++11 準拠のコードを 記述できます(互換性のために従来の Classic bcc32 コンパイラも C++Builder には搭載されています)。

この新しい BCC32C コンパイラは、エンバカデロの Web サイトから無料ダウンロードでき、コマンドラ イン コンパイラとして、(Windows のコマンドプロンプトから)C++11 準拠の C++コードをコンパイ ルできます。BCC32C コンパイラには、Dinkumware STL と構築のために必要なヘッダーインポートライ ブラリも含まれます。

BCC32C コンパイラのインストール

エンバカデロの Web サイトから BCC32C コンパイラをダウンロードします。

Combarcadero 製品 無料版 リソー	ス コミュニティ 製品の購入 求人情報	Q ¥0
C++ Compiler - FR BCC32Cコンパイラは、エンバカデロのき れているコアテクノロジーです。	EE TOOL 実績あるC++開発環境C++Builderに搭載さ	color)
登録	ログイン	
無料のC++コンパイラをダウ ウントを使ってログインする してください。 クレジットカードは不要。無料でダ ロエンバカデロ製品のユーザーである場合には、	ンロードするには、EDNアカ か、新規にアカウントを作成 ウンロードできます。 ^{こちらをチェック}	<pre>match compared and a set of the set of</pre>
姓	名	-AXX Tendo MASI Controlmantee -B Compile to .ASI (-5), then assemble to .A -Compile to .ASI (-5), then assemble to .A -D -B-Camero Advisor Assembler to use -E Specify which assembler to use -G Optimize for Size/specify use -Ol and -O2 -HoxX C Generate and use precompiled headers
		-I Set the include file search path -Jxxx Template generation options -K Set default character type to unsigned
パスワード	パスワードの確認	
会社名	電話番号	
Japan •		
私はロボットではあり よせん プライバシー・利用用的		

図 1 BCC32C コンパイラのダウンロード

ファイルを解凍すると、bin, lib, include フォルダと「Embarcadero Freeware Software License Agreement.txt」「Installing and Using the Embarcadero C++ 10.1 Berlin Command-line Compiler.txt」が 展開されます。

これらのファイルを、適当なフォルダにコピーします。今回は、「C:\data\BCC101」下にコピーしました。

環境変数の設定

BCC32C コンパイラに含まれるツールを使用するには、これらが保管されている bin フォルダに PATH を 通しておく必要があります。次のように PATH 設定に bin フォルダを追加します(この場合は、 C:\data\BCC101\bin)。

変 Dr	C:¥Program Files (x86)¥Parallels¥Parallels Tools¥Applications	新規(N)	
Pai	%SystemRoot%¥system32		
E	%SystemRoot%¥System32¥Wbem	編集(E)	
ľ	%SYSTEMROOT%¥System32¥WindowsPowerShell¥v1.0¥	会間の	
	c:¥data¥bcc101¥bin	参照(D)	
		削除(D)	
		EA (II)	
-		上八(0)	
ζ 7		下へ(0)	
变			1
Co			
		テキストの編集(T)…	
Par			
A			
R		-	
PR			
PR			`
	ОК	キャンセル	

図2環境変数の設定

BCC32C の動作チェック

コンパイラ ファイル名は bcc32c.exe ですので、それぞれコマンドラインで実行できるかチェックします。 チェックのためのフォルダは「c:\cpp_temp」とします。Windows メニューで、コマンド プロンプトを 起動したら、以下のように入力します。

mkdir cpp_temp
cd cpp_temp
bcc32c -h

bcc32c の引数に指定する -h は、コンパイラのヘルプ情報を表示するためのスイッチです。次のように表示されれば、インストールは完了です。

ש לעד אין	-	×
C:X:m/dir.opp.tomp		^
v.≠zinkutt cpp_temp		
C:¥>cd cpp_temp		
C:¥cpp_temp>bcc32c -h Embarcadero C++ 7.20 for Win32 Copyright (c) 2012-2016 Embarcadero Technologies, Inc. Available options (* = default setting, xxx = has sub-options: use -h -X): (Note: -X- or -w-XXX will usually undo whatever was set or unset by -X or -wXXX. If two options conflict, the last one specified will be used.) sysroot (Note: -X- or -w-XXX will usually undo whatever was set or unset by -X or -wXXX.		
 sysinc Set the system include file search path (if specified, sysroot will be prepended) syslib Set the system library file search path (if specified, sysroot will be prepended) savemem Set maximum SAVEMEM memory (in Mb) (This is the contiguous memory allocated to store global symbols and similar constructs. Pre-compiled headers use this memory) 		
-replacehdr Replace header name (e.greplaceHeader=a.h=b.h)		
xrtti Generate extended rtti information		
 -3 Generate 80386 protected-mode compatible instructions -4 Generate 80386/80486 protected-mode compatible instructions 		
-5 Generate Pentium instructions		
-o generate ventium pro instructions		
-B Compile to ASM (-S) then assemble to 0BJ		
-Cxxx Enable nested comments		
-D -D <name> defines 'name' as a null string or use -D<name>=<value></value></name></name>		~

図 3 bcc32c の実行

はじめての C++アプリケーションを コンパイルする

それでは、簡単な C++プログラムを作成して、BCC32C コンパイラの使い方を見ていきましょう。

Hello World アプリケーション

ここで作成するのは、プログラミング入門の定番「Hello World」です。メモ帳などのエディタを使って、 Hello World を表示するプログラムを記述します。

🦉 hello.cpp - メモ帳 \times ファイル(E) 編集(E) 書式(Q) 表示(V) ヘルプ(H) #include <vector> #include <string>
#include <iostream> int main() ł std::vector<std::string> list{"H", "e", "l", "l", "o", " ", "W", "o", "r", "l", "d"}; for (auto& str: list) { std::cout << str;</pre> } std::cout << "!" << std::endl;</pre> return 0; }

図4メモ帳でコードを記述

作成したコードは、「hello.cpp」という名称で、先ほど作成した「cpp_temp」フォルダ内に保存します。 全体のコードは次のとおりです。

#include <vector>
#include <string>
#include <iostream>

int main()

```
{
    std::vector<std::string> list{"H","e","l","o"," ","w","o","r","l","d"};
    for (auto& str: list)
    {
        std::cout << str;
    }
    std::cout << "!" << std::endl;
    return 0;
}
</pre>
```

コマンドラインで次のように、bcc32cを実行します。

bcc32c hello.cpp

プログラムに記述ミスがなければ、以下のように、コンパイル、リンクが実行されます。もし、何らかの記述ミスがあると、エラーメッセージが表示されます。

מינער איז איז איז איז דער איז גער איז גער איז גער איז גער	-	\times
C:¥cpp_temp>bcc32c hello.cpp Embarcadero C++ 7.20 for Win32 Copyright (c) 2012-2016 Embarcadero Technologies, Inc. hello.cpp: Turbo Incremental Link 6.75 Copyright (c) 1997-2016 Embarcadero Technologies, Inc. C:¥cpp_temp>		^

図 5 bcc32c の実行

フォルダ内を確認してみると、作成した hello.cpp のほかに 2 つのファイルが作成されていることが分か ります。

פני בער גער גער גער גער גער גער גער גער גער ג	-	×
		^
C:¥copp_temp>dir Li=z=z=op=dir		
トフィフ し のハリユーム ラヘルかめりません。 ガリューム シリアル表生け 0455-730D です		
C:¥cpp_temp のディレクトリ		
2017/11/28 10:41 <dir> .</dir>		
2017/11/28 10:41 <dir></dir>		
2017/11/28 10:38 256 hello.cpp		
2017/11/28 10:41 155, 136 hello. exe		
2017/11/28 10:41 131,072 hello.tds		
3 1回のファイル 280,404 ハイト		
2 個のナイレクトリー 208, 308, 317, 700 ハイトの空き領域		
C:¥cpp_temp>		

図 6 cpp_temp フォルダの内容

embarcadero

ここに作成されている hello.exe が実行可能なプログラムです。hello.exe を実行すると、次のように、 「Hello World!」と表示されます。

עדעסד אידב 🖬	-	×
C:¥cpp_temp>hello.exe Hello World! C:¥cpp_temp>		^

図7 hello.exe の実行

その他のツール

BCC32C コンパイラには、このほかにも以下のようなツールが含まれています。

名称	機能
bcc32c	Clang ベース 32bit コンパイラ
cpp32c	プリプロセッサコンパイルされる前の処理を行います
grep	ドキュメント文字列検索ツール
ilink32	32bit リンカ
implib	DLL またはモジュール定義ファイルのどちらかまたは両方を、入力として受け取
	り、インポート ライブラリ(.lib)を出力として生成します。
make	プロジェクトのコンパイルとリンクのサイクルの管理を支援するコマンドライン ユ
	ーティリティです
tdump	ファイルを構造的に分解し、ファイルの拡張子を元に出力表示形式を判断
tlib	.OBJ(オブジェクト モジュール)ファイルのライブラリを管理するユーティリティ
touch	指定されたファイルのタイムスタンプをシステムの現在日時または指定日時に変更
	します

C++Builder でもっと効率的に プログラミングする

BCC32Cは、コマンドラインツールなので、外部のエディタと組み合わせてプログラミングを行う必要が あります。コンパイル時のエラートレースやバグの発見など、コードを精査したい場合には、便利とは いえません。

統合開発環境は、このようなプログラミング時の利便性を高めるために、エディタとコンパイラとデバッガをひとつの環境にまとめて連携するようにしたものです。ここでは、BCC32Cを搭載した統合開発環境である C++Builder について説明しましょう。

C++Builder とは

C++Builder は、エンバカデロが提供する C++開発環境で、BCC32C をはじめとする複数のコンパイラが 搭載されています。C++Builder の統合開発環境では、C++アプリケーションのコーディング、ビルド、 デバッグが可能で、C++コードを効率的に作成することができます。



図 8 C++Builder の統合開発環境

embarcadero[®]

さらに、Delphi や Visual Basic と同じようにドラッグ&ドロップのビジュアル操作でユーザーインターフ ェイスやデータアクセス機能を作成できるコンポーネント指向の開発をサポートしています。そのため、 他の C++開発環境と比較して、ユーザーインターフェイスの作成にかかる工数を劇的に削減できます。

加えて、C++Builder では、Windows だけでなく、macOS、iOS、Android 向けのアプリケーションも構築できます。ひとつの C++コードから、デスクトップ、スマートフォン、タブレット、さらにはウェアラブル端末までのネイティブアプリケーションを作成できるのは大変魅力的です。



図9C++Builder のマルチデバイス開発

C++Builder には、個人やスタートアップ企業が利用できる無料の C++Builder Starter、業務アプリケーション開発に利用できる C++Builder Professional や C++Builder Enterprise といった製品ラインナップが用意されています。

C++Builder で Hello World アプリケーション作成を体験する

C++Builder が提供する基本的なプログラム作成支援機能を理解するため、先ほどコマンドライン向けに 作成した Hello World アプリケーションを、再度 C++Builder 上で作成してみましょう。ここでは、 C++Builder が得意とするビジュアル開発の要素は使いませんが、エディタ機能、プロジェクト管理機能、 デバッグ機能など、すべてのアプリケーション構築に関連する主要機能を理解するにはよい例です。

コンソールアプリケーションの新規作成

C++Builder を起動したら、メインメニューで [新規作成 | その他] を選択し、表示された「新規作成」 ダイアログで、「C++Builder プロジェクト」内の「コンソール アプリケーション」を選択します。



図 10 新規作成

embarcadero[®]

次のように、「新規コンソール アプリケーション」ウィザードが表示されるので、

新規コンソール アプリケーション	×
ソースの種類	ターゲット フレームワーク(<u>F</u>):
⊖ c(c)	なし、~
() C++(+)	☑ マルチスレッド(M) ☑ コンソール アプリケーション(A)
□ プロジェクト ソースの指定	(S)
[OK キャンセル ヘルプ(H)

図 11 「新規コンソール アプリケーション」ウィザード

「ターゲット フレームワーク」に、「なし」もしくは「ビジュアル コンポーネント ライブラリ」を指定 して、 [OK] ボタンをクリックします。

すると、次のように C++プロジェクトが作成されます。

O Project 1 - RAD Studio 10.2 - File1.cpp	– 🗆 ×
ファイル 編集 検索 表示 リファクタリング プロジェクト 実行 コンポーネント ツール ウインドウ ヘルプ 🖂 📰 🗗 칻	 デフォルトレイアクト 一番 子
構造 キメ ウエルカム ページ (Pile1.cp)	** Project1.cbproj - プロジェクトマネージャ 単 30
Щ Щ 図 ◇ Щ · · · · · · · · · · · · · · · · · ·	📅 🕶 📴 🛍 📰 💌
Includes Includes Ifpragma hdrstop Includes Inclu	
<pre>fpragma argsused fifdef wIN32 fineLude <char.h> felse typedef char TCHAR; fdefine tmain main fendif include <stdio.h> int_tmain(int argc, _TCHAR* argv[]) </stdio.h></char.h></pre>	© 種 語 三 · 圖 圖 · 종 · ファイル ProjectGroup1 中国 ProjectLoxe 中國 ProjectLoxe 中國 ProjectIQC000 中國 ProjectIPCH1.h
オブジセウト インスペクタ **	Project1.c モデル ビュー テータ エクス マルチデパイ ツール パレット キョ ロート目 ● (日本) ロート ● (日本)
▶ ● ■ 1:1 挿入 (空事あり) □→下 履歴	⊡ Delphi プロジェクト
₩v#r=%	
<mark>愛</mark> =	

図 12 新規 C++プロジェクト

プロジェクトマネージャ

画面右側にあるプロジェクトマネージャには、作成した C++プロジェクト(デフォルトで「Project1」の ように名前が付けられています)を構成する要素が表示されています。



図 13 プロジェクトマネージャ

プロジェクトマネージャには、アプリケーションを作成するために必要となるファイルを追加、管理で きるほか、ビルド構成やターゲットプラットフォームを設定できます。

この例では、「Win32」がターゲットプラットフォームで、「Debug」ビルド(デバッガによるトレース ができるようにデバッグ情報が含まれるビルド)が行われます。C++Builder では、ターゲットを 「Win64」や「macOS」などに変更することも可能です(エディションによって異なります)。

C++Builder のプロジェクトには、以下のようなファイルが含まれます。今回作成した、コンソール アプ リケーションでは、以下のようなファイルによって構成されています。

ファイル名	詳細
[プロジェクトファイル名]. cbproj	最終アウトプットされる EXE ファイル名
[C++ソースファイル名].cpp	main 関数が入る C++ソースファイル
[プリコンパイルヘッダ] PCH1.h	プリコンパイルヘッダファイル

プロジェクト名は、作成される実行ファイル(EXE ファイル)の名称となります。

エディタのコーディング支援機能

先ほどメモ帳に記述したのと同じコードを入力してみましょう。C++Builder のコードエディタには、構 文強調表示機能が搭載されており、コードを構成する予約語や文字列リテラルなどを、視認しやすくし てくれます。

また、複数取り消し機能、コード補完機能、言語要素の状況依存型ヘルプなども備えており、効率的な コーディング作業が可能です。以下は、コード補完機能を使用しているところです。コード補完機能は、 現在のカーソル位置で使用できるシンボルのドロップダウンリストを表示します。



図 14 コード補完機能

特定の言語のコード補完を呼び出すには、コードエディタを使用している状態で、Ctrl+Space を押します。



コーディング中にメソッドを記述していくと、呼び出すメソッドの引数の名前と型をヒントとして表示 します。

ウエルカム ページ 😓 main_program.cpp
🧕 🕶 🚽 _tmain 🗸
#pragma hdrstop
<pre>#pragma argsused</pre>
and the second se
#ifdef_WIN32
- #include <tchar.h></tchar.h>
Telse
typeder char_ICHAR;
fendif
10
<pre>include <iostream></iostream></pre>
15 std::wcscpy()
<pre>return 0; wchar_t *_dst, const wchar_t *_src</pre>
1

図 15 引数のヒント表示

構造強調表示では、コードブロックをグループ化し、該当するキーワードを識別します。次のようにネ ストしたブロックでは、交互に変わる4色を使用して、特定のブロックの深さを区別します。

ウェルカム ページ 😸 main_program.cpp	
🧃 ≠ 🖶 ≠ _tmain 🗸	•
<pre>#pragma hdrstop</pre>	
<pre>#pragma argsused</pre>	
finely wind:	
felse	
typedef char TCHAR;	
fdefine tmain main	
+ #endif	
10	
<pre>#include <iostream></iostream></pre>	
#include <vector></vector>	
Dint tmain(int args TCHAD& args[1)	
<pre>std::vector<int> vil{0,1,2,3};</int></pre>	
if (vil[0] == vil[1])	
· •	
if (vil[2] == vil[3]) {	
20 while (1)	
for (int $i = 0; i < 10; i++)$	
[101 (110 1 - 0, 1 < 10, 1++)]	
30 return 0:)	

図 16 構文強調表示

コンパイルエラーを調べる

コードを入力したらプログラムをビルドしてみましょう。C++Builder では、 [実行] メニューを選択す るだけで、必要なファイルをコンパイルして、アプリケーションを作成、実行します。

ビルドを実行すると、次のようにコンパイラメッセージが表示されます。



図 17 コンパイル実行

上記の図では、残念ながらコードの記述にミスがあり、コンパイルエラーになってしまったようです。 BCC32Cコンパイラを使っていたときは、コンパイルエラーが発生したら、エラーメッセージをひとつず つ確認しながら、エディタで該当する行を探さなければなりませんでした。

C++Builder では、エラーメッセージが表示されるメッセージウィンドウとエディタが連動するので、エ ラーメッセージを確認するときに、同時に該当するコードを見つけることができます。 次の例では、式のうしろにセミコロン「;」の入力を忘れてエラーになってしまいました。エディタの該 当する行が赤く表示されているので、すぐに修正すべき箇所を発見できます。

🤓 project1 - RAD Studio 10.2 - main_program.cpp	- 🗆 ×
ファイル 編集 検索 表示 リファクタリング プロシェクト 実行 コンポーネント ツール ウインドウ ヘルプ 🖂 🛅 🗊 🌗	
Image: State in the state	* ** project1.cbproj - 方522/b マネーシャ * ** **
	C:¥Embarcadero¥test1111¥Console¥project1.cbproj project1.c モデル ビュー「テータ エクス 「マルチデノイ ツール パレット ・★★ ① ~ 】 ○ 検索 ② Delphi プロジェクト Delphi ファイル ~ ③ C++Builder プロジェクト C++Builder ファイル ③ Delphi プロジェクト
	□ Delphi フロジェクト I マルチデバイス プロジェクト ×
project1.cbprojをコンパイル中 (Debug, Win32) B bc32c の ¹ main_program.cpp(コマルド タイン [bc32c 15-] main_program.cpp(24): 式の後には '; が必要です [bc32c 智力] main_program.cpp(24): 式の後には '; が必要です [bc32c 智力] main_program.cpp(24): 式の後には '; が必要です [bc32c 智力] main_program.cpp(24): 式の後には '; が必要です [bc32c 15-] main_program.cpp(24): 式の後には '; がの [bc32c 15-] main_program.cpp(24): 式の(25-] main_program.cpp(24): 30-] main_program.cpp(24): 30-	

図18コンパイルエラー箇所の表示

デバッガを使う

エラー箇所を修正したら、再びプログラムをビルド/実行します。プログラム実行時には、ソースコー ドの特定箇所で、プログラムを一時停止させる「ブレークポイント」を設定できます。

ブレークポイントを設定するには、停止させたい行のソースコードの左余白をクリックします。ブレー クポイントでプログラムを停止させることで、変数の内容を確認したり、1 行ずつ実行して動きを確認す ることができます。



🥝 Project 1 - RAD Studio 10.2 - File1.cpp [停止 - スレッド 9400]																-	□ ×				
ファイル 編集	検索	表示	リファクタリング	プロジェク	実行	コンポーネント	ツール	ウィンドウ	ヘルプ		🛅 🔊 🤊				e 7	テバッグ レイアウ	٢	- 6 6	の検索		1
	1 😁 💕	- 8 8	1 To 10		- II 🖬	1 G (= G	32 6	/ 양ト Windows	~		~ 6	•		0							1
呼び出し履歴 - スレッ	F 9400		# X	ウェルカム ペー	🔊 🍃 File1.c	pp									v 3	Project	1.cbproj ·	・プロジェクト	マネージャ	# 3	a
:0040172E mai	n(argc=1, ar	gv=:0222A	(680)	💐 + 🖶 •										- 🔎		*7 -	📭 🛍	100 V			1
 :0041ffa8 ;s :77=2592d ptdl 	tartup I RtiCotAppC	ontainorNa	medObject		endif											^	- 1 -		-		-
 :77a257fd ntdll 	.RtlGetAppCo	ontainerNa	medObject	10	include <	vector>										G 4	2= 1=	- 🗟 🕸	- ⇒ -		
				1 1	include <	string>										ファイル	abConvert				
8548-1 89 71	0.400			1	include <:	iostream>										Proje	ctGroup1	٥			
監視式一見・スレット 監視式	(書		+×	• - <mark>-</mark> -1	nt tmain	(int argc,	TCHAR*	argv[])								÷-**	ビルド構成	(Debug)			
p- ☑ str	{ 4294967	7295 }	^		-	-		_									ターゲットフ	ラットフォーム (Win32)		
- npos	42949672	95 (0xFFFF	FFFF)	•	std::ve	ector <std::s< td=""><td>string> :</td><td>List{"H","</td><td>e", "1", "1</td><td>1","0",</td><td>","W",</td><td>"o", "r"</td><td>"1","d"</td><td>};</td><td></td><td>÷.</td><td>- 32 Ey</td><td>Windows</td><td></td><td></td><td></td></std::s<>	string> :	List{"H","	e", "1", "1	1","0",	","W",	"o", "r"	"1","d"	};		÷.	- 32 Ey	Windows			
BMypair	{ { { { 'H¥0	0¥0¥0¥0¥0¥0 0¥0¥0¥0¥0¥0	#0¥0¥ ¥0¥0¥	• . <mark>.</mark>	(1004 501. 11											File1.cpp	CU1 h			
ian _nyvaiz	{ "H¥0¥0¥	40¥0¥0¥0¥0¥0	¥0¥0¥0	20	st	i::cout << s	tr;										Projectip	CHI.n			
My	1 (0x0000	0001)		📫 21	}	out << "!!" <	< etd	andl.													
My	. 15 (0x000	0000F)		• • •	return 0;)																
	1			• <u> </u> 1																	
🗄 🗹 argv	:0222A68	0		-																	
é- [0]	:022319B	0 "¥¥¥¥Mac	¥¥Ho																		
	** 92 (0) '##' 92 (0)	(SC) (SC)																			
	'M' 77 (0x	4D)		<											>	~					
[3]	'a' 97 (0x6	51)	*	• • •	21: 1	挿入	変更あり	a−۴ 🕅								_					
「監視式」							50,0007														1
ZLAVE ID	採湯		17 TR	位業			2	は換チェーン												+ 2	5
Project1.exe (5	260)		- Car																		1
9400	停止		ステップ	¥¥Ma	*Home¥Doo	uments¥Embar	cade														1
Pb 6484	停止		不明	\$77A	396C																1
ブレークポイント一覧																				# 3	¢
•+ • •× 🔏 '	🤹 🗞 🍕																				1
ファイル名/アドレス		行/長さ	状態			スレッド		アクシ	(a)		1	スカウント	グループ								1
File1.cop		20						ブレー	-9 -9												1
									-												
イベント ロガ ブレーク	ポイントー覧																				-
	36 121 26																				_

図 19 デバッガでプログラムを停止

C++Builderの製品情報、ならびに無料トライアル版、無料の Starter Edition のダウンロードいついては、 以下のページをご覧ください。

https://www.embarcadero.com/jp/products/cbuilder

Embarcadero、Embarcadero Technologies ロゴならびにすべてのエンバカデロ・テクノロジーズ製品またはサービ ス名は、Embarcadero Technologies, Inc.の商標または登録商標です。その他の商標はその所有者に帰属します。