



3DS

3DS プログラミングマニュアル

アプリケーション作成フロー編

2015-11-05
Version 1.3

Nintendo Confidential

本ドキュメントの内容は、機密情報であるため、厳重な取り扱い、管理を行ってください。
任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。

This document contains confidential and proprietary information of Nintendo and is also protected under the copyright laws of the United States and foreign countries.

No part of this document may be released, distributed, transmitted or reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without permission in writing from Nintendo.

© 2015 Nintendo Co., Ltd. All rights reserved.

記載されている会社名、製品名等は、各社の登録商標または商標です。

目次

1. はじめに	6
1.1. 本書の読み方	6
1.2. 用語解説	6
2. アプリケーションの種類	8
2.1. カードアプリ	8
2.2. ダウンロードアプリ	8
2.2.1. インポート	9
2.2.1.1. 上書きインポート	9
2.2.2. あらかじめダウンロード	10
2.2.3. 体験版について	10
2.2.4. カードアプリ/Wii ウェア/DSi ウェアとの違い	10
2.2.5. TWL の NAND アプリのインポート	10
2.3. 子機プログラム	10
3. 実装のヒント	12
3.1. 起動前の初期化处理	12
3.2. エントリ関数	12
3.2.1. 特殊な操作のハンドリング	12
4. ビルド前の準備	13
4.1. RSF ファイル	13
4.1.1. 必須の設定	13
4.1.2. セーブデータを使用する場合	14
4.1.2.1. ゲームカード種別が Card1 のカードアプリ	14
4.1.2.2. ダウンロードアプリ	14
4.1.2.3. ゲームカード種別が Card2 のカードアプリ	15
4.1.2.4. 体験版のセーブデータにアクセスする場合	15
4.1.2.5. ほかのアプリケーションのセーブデータにアクセスする場合	16
4.1.3. 拡張セーブデータを使用する場合	16
4.1.4. マスター提出用 ROM 作成時の設定	17
4.1.5. リマスターバージョンの設定	18
4.2. BSF ファイル	19
4.3. 電子取扱説明書	20
5. ビルド	21
5.1. CTR-SDK のビルドシステムでビルドする場合	21
5.1.1. カードアプリの作成	22
5.1.2. ダウンロードアプリの作成	23
5.1.3. 子機プログラムの作成	23
5.2. CTR-SDK のビルドシステム以外でビルドする場合	24
5.2.1. AXF ファイルの作成	24

5.2.2. ICN ファイルと BNR ファイルの作成	25
5.2.3. 電子取扱説明書の CFA ファイルの作成	25
5.2.4. カードアプリの作成	25
5.2.4.1. CCI ファイルの作成	25
5.2.5. ダウンロードアプリの作成	26
5.2.5.1. CXI ファイルの作成	26
5.2.5.2. CIA ファイルの作成	27
5.2.6. 子機プログラムの作成	27
5.2.6.1. CXI ファイルの作成	27
5.2.6.2. CIA ファイルの作成	28
5.2.6.3. CFA ファイルの作成	28
6. デバッグ	29
6.1. 起動時に表示されるメニューによって異なるデバッグ方法	29
6.1.1. 開発用メニューから起動する場合	29
6.1.2. HOMEメニューから起動する場合	29
6.2. cia ファイルをデバッガソフトウェアで実行する場合	30
7. DLL (動的モジュール) の利用	31
7.1. 実装	31
7.2. ビルド前の準備	31
7.3. ビルド	31
7.3.1. CTR-SDK のビルドシステムでビルドする場合	31
7.3.2. CTR-SDK のビルドシステム以外でビルドする場合	33
7.4. デバッグ	33
8. ダウンロード販売可能なカードアプリの開発手順	34
8.1. 新規に開発するカードアプリを併売対応するケース	34
8.1.1. 企画段階	34
8.1.2. 取得する製品コードとユニーク ID	34
8.1.3. 電子取扱説明書	34
8.1.4. セーブデータ、追加データ	35
8.1.5. セーブデータの巻き戻しについて	35
8.1.6. ゲーム内の表記、表示	35
8.1.7. ダウンロードプレイ	36
8.1.8. メディアへのアクセス速度	37
8.1.9. アプリバイナリの生成 (ビルド)	37
8.1.10. デバッグ	37
8.1.10.1. セーブデータの流用	37
8.1.11. マスター提出	38
8.2. 発売済みのカードアプリをダウンロードアプリとして販売するケース	38
8.2.1. SDK バージョン	38
8.2.2. 電子取扱説明書	38

8.2.3. セーブデータの互換性について	39
8.2.4. セーブデータの巻き戻しについて	39
8.2.5. ゲーム内の表記、表示	39
8.2.6. ダウンロードプレイ	39
8.2.7. 動作確認	39
8.2.8. マスター提出	39
8.3. アプリケーションの修正が必要になった場合	40
8.3.1. 配信開始後に修正が必要になった場合	40
8.3.2. 発売済み、かつパッチをリリース済みのカードアプリを配信する場合	41
更新履歴	42

コード

コード 4-1. RSF ファイルの記述例(必須の設定)	14
コード 4-2. RSF ファイルの記述例(Card1 のカードアプリでセーブデータを使用する場合)	14
コード 4-3. RSF ファイルの記述例(ダウンロードアプリでセーブデータを使用する場合)	15
コード 4-4. RSF ファイルの記述例(Card2 のカードアプリでセーブデータを使用する場合)	15
コード 4-5. RSF ファイルの記述例(拡張セーブデータ)	17
コード 4-6. RSF ファイルの記述例(マスター ROM 提出時)	18
コード 4-7. RSF ファイルの記述例(リマスターバージョン)	18
コード 4-8. BSF ファイルの記述例	19
コード 5-1. OMakefile の記述例(カードアプリ)	22
コード 5-2. OMakefile の記述例(ダウンロードアプリ)	23
コード 5-3. OMakefile の記述例(子機プログラム)	24
コード 7-1. 動的モジュールの OMakefile の例	32
コード 7-2. アプリケーションの OMakefile に追加する設定の例	32
コード 7-3. 動的モジュールのファイルの格納を自動化する OMakefile の例	32

表

表 1-1. 用語一覧	6
表 2-1. カードアプリの特徴	8
表 2-2. ダウンロードアプリの特徴	8
表 2-3. 子機プログラムの特徴	11
表 4-1. 必須の設定	13
表 4-2. セーブデータを使用する場合の設定(Card1 のカードアプリ)	14
表 4-3. セーブデータを使用する場合の設定(ダウンロードアプリ)	15
表 4-4. セーブデータを使用する場合の設定(Card2 のカードアプリ)	15
表 4-5. 体験版のセーブデータにアクセスする場合の設定	15
表 4-6. ほかのアプリケーションのセーブデータにアクセスする場合の設定	16

表 4-7. 拡張セーブデータを使用する場合の設定	16
表 4-8. マスター提出用 ROM 作成時の設定	17
表 4-9. リマスターバージョンの設定	18
表 4-10. ダウンロードアプリのバージョンアップ時に変更が禁止されている項目	18
表 5-1. カードアプリをビルドする OMakefile に記述する内容	22
表 5-2. ダウンロードアプリをビルドする OMakefile に記述する内容	23
表 5-3. 子機プログラムをビルドする OMakefile に記述する内容	24
表 6-1. デバッグに使用するハードウェアとデバッグソフトウェアの組み合わせ	29
表 7-1. 動的モジュールのビルドで追加された設定	31
表 8-1. 電子取扱説明書の新規作成や改訂の要・不要	38
表 8-2. マスター提出時に必要な提出物と付与すべきバージョン	40



図 5-1. アプリケーションをビルドするまでの概略	21
----------------------------	----

1. はじめに

本ドキュメントは、3DS で動作するアプリケーションの作成に関連する情報をまとめたもので、アプリケーション開発者を対象としています。

1.1. 本書の読み方

アプリケーションの作成に関連する情報の紹介を主な目的としているため、詳細な情報については、関連するドキュメントを適宜参照してください。

以降の章の構成は以下のようになっています。

「2. アプリケーションの種類」では、3DS で作成可能なアプリケーションの種類と、それぞれの特徴を説明しています。

「3. 実装のヒント」、「4. ビルド前の準備」、「5. ビルド」では、アプリケーション作成の基本手順を説明しています。

「6. デバッグ」では、アプリケーションのデバッグ方法を説明しています。

「7. DLL(動的モジュール)の利用」では、アプリケーションで動的モジュールを利用する場合に必要な手順を説明しています。

「8. ダウンロード販売可能なカードアプリの開発手順」では、同じアプリケーションをパッケージ版とダウンロード版の 2 つの販路で販売する場合に必要な手順を説明しています。

1.2. 用語解説

本ドキュメントで使用される用語を簡単に解説します。これらの用語は一般的な意味とは異なる場合があります。

表 1-1. 用語一覧

用語	説明
カードアプリ	ゲームカードに格納され、実行される 3DS で動作するアプリケーション。実行時にはゲームカードからデータを読み込みます。
ダウンロードアプリ	SD カードに格納され、実行される 3DS で動作するアプリケーション。実行時には SD カードからデータを読み込みます。
CXI (CTR eExecutable Image)	アプリケーションの実行イメージ形式。また、それを格納したファイル、ファイルにつけられる拡張子。
電子取扱説明書	3DS の画面で表示されるアプリケーションの取扱説明書。CTR-ManualTools で作成します。
CFA (CTR File Archive)	データコンテナとして扱われるバイナリ形式。またそれを格納したファイル、ファイルにつけられる拡張子。
CCI (CTR Card Image)	カードアプリの最終 ROM イメージ形式。また、それを格納したファイル、ファイルにつけられる拡張子。内部に CXI を含んでいます。
CIA (CTR Importable Archive)	ダウンロードアプリの最終 ROM イメージ形式。また、それを格納したファイル、ファイルにつけられる拡張子。内部に CXI を含んでいます。ダウンロードプレイで配信する子機プログラムもこの形式で作成します。
RSF (Rom Spec File)	CXI を作成するときに使用する各種パラメータを記述したファイル。
リマスターバージョン	RSF に記述され CXI 内部に格納されるアプリケーションのバージョン番号。

インポート	ダウンロードアプリを SD カードに格納し、起動可能な状態にすること。
HOMEメニュー	3DS を起動したときに表示される画面。起動するアプリケーションを選択するために用いられます。
開発用メニュー	PARTNER-CTR Debugger や開発実機を起動したときに表示される開発用の画面。HOMEメニューの代わりとして、最小限の機能を有しています。
DevMenu	SD カード上の CIA ファイルのインポート、拡張セーブデータの表示などの機能を持つ、開発用のアプリケーション。
アーカイブ	CTR-SDK の FS ライブラリが操作の対象とする個々のファイルシステム。Windows PC でいう「ドライブ」に相当します。
セーブデータ領域	セーブデータを格納するために確保される領域。ダウンロードアプリでは SD カード上に確保され、作成した本体でのみ使用可能です。
拡張セーブデータ領域	セーブデータとは別に作成・管理される SD カード上のデータ領域。作成した本体でのみ使用可能です。領域に作成された拡張セーブデータは「追加データ」とも呼ばれます。
モジュール	実行コードを意味のある単位でまとめたもの。
動的モジュール	動的にメモリに読み込み、実行することのできるモジュール。
静的モジュール	最初に実行される、アプリケーションの本体となるモジュール。動的モジュールを用いていないアプリケーションは、静的モジュールだけで構成されているとも言えます。
DLL	動的モジュールを実現する仕組み。また、動的モジュールそのもの。
CRS	静的モジュールの情報。また、それを格納したファイル、ファイルにつけられる拡張子。
CRO	動的モジュールのファイル形式。また、それを格納したファイル、ファイルにつけられる拡張子。
CRR	動的モジュールの登録情報。また、それを格納したファイル、ファイルにつけられる拡張子。CRO の登録情報が 1 つ以上格納されています。
パッケージ版	併売されているアプリケーションのうち、ゲームカードに記録され、店頭で販売されているものを指します。
ダウンロード版	併売されているアプリケーションのうち、ダウンロード販売されているものを指します。
改訂版	一度アプリケーションをリリースしたのちに、何らかの変更を施してリリースされる同タイトルの新しい版のアプリケーション。
パッチ	アプリケーションのいくつかの要素を置き換えて実行する仕組み、またはその置き換えに使用するデータ。
標準アプリ	SNAKE で実行した際に標準モードで動作するアプリケーション。発売済みを含む、ニンテンドー3DS専用ソフトがこれにあたります。
拡張アプリ	SNAKE で実行した際に拡張モードで動作するアプリケーション。CTR で実行することもできます。CTR で動作した際の挙動は、SNAKE の標準モードで動作したときと同等になります。

2. アプリケーションの種類

3DS で開発することのできるアプリケーションの種類には、以下のものがあります。

2.1. カードアプリ

ROM とセーブデータが一体化したゲームカードに書き込まれるアプリケーションです。店舗などで販売されているパッケージソフトがこれにあたります。

アプリケーションの保存場所とサイズ、セーブデータ、拡張セーブデータ、電子取扱説明書への対応については、以下のようになっています。

表 2-1. カードアプリの特徴

項目	内容
アプリケーションの保存場所	ゲームカード (ニンテンドー 3DS カード)
アプリケーションのサイズ	Card1: 128 MByte～4 GByte (実際に使用できるサイズは異なります) Card2: 512 MByte～2 GByte (セーブデータと共用)
セーブデータの保存場所	ゲームカードのバックアップ領域 (ROM 領域とは別の領域です)
セーブデータのサイズ	Card1: 128 KByte または 512 KByte (実際に使用できるサイズは異なります) Card2: 1 MByte 単位で指定可能 (カード容量の半分まで)
拡張セーブデータ	使用可能 (追加データ管理画面で削除することができます)
電子取扱説明書への対応	必須

注意: Card2 を使用する場合は、事前に弊社までご相談ください。

2.2. ダウンロードアプリ

SD カードにインポートされるアプリケーションです。ニンテンドー e ショップで販売されているソフトやいつの間にも通信でダウンロードされるソフトがこれにあたります。

アプリケーションの保存場所とサイズ、セーブデータ、拡張セーブデータ、電子取扱説明書への対応については、以下のようになっています。

表 2-2. ダウンロードアプリの特徴

項目	内容
アプリケーションの保存場所	SD カード
アプリケーションのサイズ	カードアプリをダウンロード販売にも対応する場合は、最大 4 GB です。 ダウンロード販売のみの場合は、基本的に 512 MB までです。(512 MB 以上のサイズが必要な場合は弊社窓口までご連絡ください)
セーブデータの保存場所	SD カード (インポート時にセーブデータ領域が確保されます。ダウンロードアプリを削除したときに一緒に削除され、セーブデータのみを削除することはできません)

セーブデータのサイズ	128 KByte または 512 KByte (実際に使用できるサイズは異なります。512 KByte 以上のサイズが必要な場合は弊社窓口までご連絡ください)
拡張セーブデータ	使用可能 (ダウンロードアプリを削除しても削除されません。追加データ管理画面で削除することができます)
電子取扱説明書への対応	必須

注意: ダウンロードアプリ (cia) としてマスター提出した場合は、後からカードアプリとして販売を依頼することが出来ません。併売をご検討の場合はカードアプリ (cci) としてマスター提出するようにお願いします。

2.2.1. インポート

ダウンロードアプリを実行するには、まず SD カードにアプリケーションをインポートする必要があります。

製品版では、ニンテンドーeショップなどからのダウンロードが完了した時点でインポートされています。

開発時のインポートには、DevMenu を使用する方法と PARTNER-CTR (デバッグソフトウェア) または IS-CTR-DEBUGGER (デバッグソフトウェア) を使用する方法があります。

DevMenu でインポートするには、CIA ファイルを SD カードに書き込んでおく必要があります。

PARTNER-CTR によるインポートは、CIA ファイルを読み込む (ドラッグアンドドロップも可) か、「IMPORT」コマンドを実行することで行われます。

IS-CTR-DEBUGGER によるインポートは、CIA ファイルを読み込む (ドラッグアンドドロップも可) か、「IMPORT」コマンドを実行することで行われます。

補足: DevMenu の操作については、CTR-SDK の CTR ツール「DevMenu」のリファレンスを参照してください。

PARTNER-CTR のコマンドの詳細については、「PARTNER-CTR デバッグマニュアル」を参照してください。

IS-CTR-DEBUGGER のコマンドの詳細については、IS-CTR-DEBUGGER のヘルプを参照してください。

2.2.1.1. 上書きインポート

インポートされたダウンロードアプリはユニーク ID で識別されます。そのため、ユニーク ID が同じダウンロードアプリがインポート済みの場合、新たにインポートするダウンロードアプリのバージョンが同じまたは古い場合はインポートされず、新しい場合は上書きインポートが行われます。

上書きインポートが行われる場合、通常はセーブデータの内容が維持されます。そのため、不具合が起こらないように、セーブデータのバージョンなどでしっかりと管理する必要があります。また、インポートするバージョンとインポートされているバージョンとでセーブデータ領域のサイズが異なる場合、すべてのセーブデータ領域の内容がクリアされます。

注意: 製品版のバージョンを変更した際にセーブデータサイズを変更することは禁止されています。

デバッグ時に古いバージョンのダウンロードアプリをインポートする場合や、セーブデータ領域の内容をクリアする必要がある場合は、DevMenu や本体設定のソフト管理画面で、インポート済みのダウンロードアプリを削除してからインポートしてください。ダウンロードアプリを削除しても、拡張セーブデータは削除されないことに注意してください。

2.2.2. あらかじめダウンロード

ダウンロードアプリは、あらかじめダウンロード機能に対応することで、発売日より前にニンテンドーeショップで購入やダウンロードができるようになります。

詳細は「3DS オーバービュー」の「あらかじめダウンロード」を参照してください。

2.2.3. 体験版について

ソフトの体験版をダウンロードアプリで開発することができます。

体験版と製品版が同じユニーク ID であっても、別のダウンロードアプリとしてインポートされます。そのため、体験版から製品版へのセーブデータ自体の移行はできませんが、拡張セーブデータの共有を利用して移行することはできます。

拡張セーブデータを共有する場合、製品版を起動したあとで体験版を起動すると、製品版で更新された拡張セーブデータの内容で体験版が起動することになりますので注意が必要です。

体験版を削除した場合、体験版のセーブデータは削除されますが、拡張セーブデータは削除されません。

補足: 体験版の詳細については、「ニンテンドー3DS 体験版作成手順書 - ダウンロード型」を参照してください。

2.2.4. カードアプリ/Wii ウェア/DSi ウェアとの違い

「3DS オーバービュー」の「ダウンロードアプリ」を参照してください。

2.2.5. TWL の NAND アプリのインポート

ctr_makecia の `-srl` オプションを使用すると、TWL の NAND アプリとしてビルドした SRL ファイルから CIA ファイルを作成することができます。作成した CIA ファイルは、ダウンロードアプリと同様に SD カードにインポートすることができます。

詳しくは CTR-SDK のツール「ctr_makecia」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makecia32.exe -srl MyTwlApp.srl -o MyTwlApp.cia
```

注意: SRL ファイルから作成された CIA ファイルは 3DS での DSi ウェアの動作チェックに使用します。インポート後は PARTNER-CTR でも実行できますが、デバッグできません。

2.3. 子機プログラム

ダウンロードプレイで親機から子機に配信されるアプリケーションです。

アプリケーションの保存場所とサイズ、セーブデータ、拡張セーブデータ、電子取扱説明書への対応については、以下のようになっています。

表 2-3. 子機プログラムの特徴

項目	内容
アプリケーションの保存場所	本体 NAND 領域 (1 つのみ保存され、異なるプログラムを受信したときは上書きされます。同じバージョンの同じプログラムは再受信しません)
アプリケーションのサイズ	(展開後のサイズで) 32 MByte まで
セーブデータ	使用不可
拡張セーブデータ	使用可能 (子機プログラムが上書きされても削除されません。追加データ管理画面で削除することができます)
電子取扱説明書への対応	不要
備考	HOMEメニューには表示されないため、バナー (モデルとサウンド) は必要ありません。

補足: ダウンロードプレイのデバッグ用に、同じバージョンの同じプログラムであっても強制的に受信するように Config ツールで設定可能です。

注意: ダウンロードアプリがダウンロードプレイのサーバになることはできませんが、現時点では、接続してきた子機のシステムアップデートを行うことができません。親機で動作しているアプリケーションがダウンロードアプリの場合、システムアップデートが必要な子機からの接続は警告なしに拒否されます。そのため、ダウンロードアプリがダウンロードプレイを開始する際に、弊社が用意するメッセージをアプリケーションで表示しなければなりません。

詳しくは、DLP ライブラリの関数リファレンスを参照してください。

3. 実装のヒント

この章では、アプリケーションの実装に役立つ情報や注意点などを説明します。

3.1. 起動前の初期化処理

CTR-SDK の「システムプログラミングガイド」に記載されているように、アプリケーションが起動するまでに行われる初期化処理のうち、以下の処理についてはアプリケーション独自の実装が可能となっています。

- `nninitStartup()` の定義
- C++ スタティックイニシャライザ
- C スタティックイニシャライザ

このうち、`nninitStartup()` の定義では、スタティックイニシャライザの前に行う、メモリ管理機構の初期化処理を実装します。この関数は `weak` シンボルで定義されているため、同名の関数をアプリケーションで定義すると、アプリケーション独自の実装に置き換えることができます。

デフォルトの実装は `$CTRSDK_ROOT/sources/libraries/init/init_Default.cpp` の `nninitStartupDefault()` にありますが、`nninitStartup()` を独自の処理に置き換えることが強く推奨されています。

3.2. エントリ関数

初期化処理のあとで呼び出されるエントリ関数(プログラムのメイン関数)は `nnMain()` です。アプリケーションの処理はこの関数で実装します。

`nnMain()` から抜けるとアプリケーションを終了し、HOMEメニュー(もしくは開発用メニュー)に戻ります。

3.2.1. 特殊な操作のハンドリング

アプリケーションは以下の操作が行われたときの動作をハンドリングしなければなりません。

- HOMEボタンが押されたとき
- 蓋が閉じられ、スリープ状態に移行するとき
- 電源ボタンが押されたとき

詳しくは、「3DS プログラミングマニュアル – システム編」を参照してください。

4. ビルド前の準備

この節では、プログラム以外で、アプリケーションのビルドまでに用意しておかなければならないファイルの作成方法を説明します。

ビルドまでに、以下のファイルを用意する必要があります。

- RSF ファイル
- BSF ファイル
- 電子取扱説明書

4.1. RSF ファイル

RSF ファイルは ctr_makerom で CCI ファイルや CXI ファイルを作成する際に必要となる設定ファイルです。

詳しくは CTR-SDK のツール「ctr_makerom」のリファレンスを参照してください。

4.1.1. 必須の設定

以下の項目はアプリケーションを作成する上で必須となる項目です。

表 4-1. 必須の設定

大項目	小項目	設定
TitleInfo	Category	アプリケーションの種類に従って以下のように指定します。 <ul style="list-style-type: none">● カードアプリ:「Application」● ダウンロードアプリ:「Application」(体験版は「Demo」)● 子機プログラム:「DlpChild」● 追加コンテンツ:「AddOnContents」 ※ 省略した場合は、ビルド時の DESC ファイルによってアプリケーションの種類が決定されます。
	UniqueId	弊社発行のユニーク ID を指定します。 ソフトの実験、試作用に試作用コード(0xFF000~0xFF3FF)を指定することができますが、複数のアプリケーションで重複しないように注意してください。 ダウンロードプレイに対応する場合は、親機と子機で同じユニーク ID を指定しなければなりません。
	ChildIndex	ダウンロードプレイで配信する子機プログラムを作成する際に、そのインデックス(0~255)を指定します。
	DemoIndex	ダウンロードアプリ型体験版を作成する際に、そのインデックス(1~255)を指定します。詳細については「ニンテンドー3DS 体験版作成手順書 - ダウンロード型」を参照してください。
	Variation	追加コンテンツを作成する際に、そのインデックス(0~255)を指定します。
Rom	HostRoot	ROM アーカイブに含めるファイルが格納されているディレクトリを指定します。 指定されたディレクトリをルートディレクトリに、配下のファイルとディレクトリ構造がそのまま ROM アーカイブに格納されます。 OMakefile の ROMFS_ROOT を参照する場合は、「\$(ROMFS_ROOT)」を指定してください。

	Reject	ROM アーカイブから除外するファイルやディレクトリを指定します。 除外するファイルはワイルドカードまたは正規表現で指定します。
	SaveDataSize	セーブデータ領域のサイズを nKB/nMB/nGB の形式で指定します。 省略した場合は 0KB が指定されます。
CardInfo	MediaType	カードアプリを作成する際に、ゲームカードの種別 (Card1/Card2) を 指定します。 セーブデータ領域のサイズが 512 KB を超えるダウンロードアプリは 「Card2」を指定してください。 省略した場合は「Card1」が指定されます。

コード 4-1. RSF ファイルの記述例 (必須の設定)

```
TitleInfo:
  Category: Application
  UniqueId: 0xFF3FF
Rom:
  HostRoot: $(ROMFS_ROOT)
```

4.1.2. セーブデータを使用する場合

アプリケーションでセーブデータを使用する場合、アプリケーションの種類によって設定すべき項目が異なります。

補足: セーブデータ領域内にはファイルシステムの管理領域などが含まれるため、アプリケーションで実際に
使用できる領域は指定した値より小さくなります。セーブデータ領域サイズごとの、実際に利用可能な
サイズの値については関数リファレンスの「セーブデータファイルシステム容量計算シート」を参照して
ください。

なお、`nn::fs::GetArchiveFreeBytes()` を呼び出すことで、アプリケーションの実行中にアー
カイブの空き容量を取得することができます。

4.1.2.1. ゲームカード種別が Card1 のカードアプリ

以下の項目を設定してください。

表 4-2. セーブデータを使用する場合の設定 (Card1 のカードアプリ)

大項目	小項目	設定
Rom	SaveDataSize	「0KB」、「128KB」、「512KB」から指定します。

コード 4-2. RSF ファイルの記述例 (Card1 のカードアプリでセーブデータを使用する場合)

```
Rom:
  SaveDataSize: 512KB
```

4.1.2.2. ダウンロードアプリ

以下の項目を設定してください。

表 4-3. セーブデータを使用する場合の設定(ダウンロードアプリ)

大項目	小項目	設定
Rom	SaveDataSize	Media Type の指定により異なります。 Card1 の場合:「0KB」、「128KB」、「512KB」から指定します。 Card2 の場合:「nMB」または「nGB」の形式で指定します。 1 MByte 単位でのみ指定が可能で、50 MByte を超える場合は事前に弊社までご相談ください。 なお、ダウンロード専売の場合は 512 KByte まで、カードアプリと併売する場合は 20 MByte までにすることを推奨します。

コード 4-3. RSF ファイルの記述例(ダウンロードアプリでセーブデータを使用する場合)

```
Rom:
  SaveDataSize: 512KB
```

4.1.2.3. ゲームカード種別が Card2 のカードアプリ

以下の項目を設定してください。

表 4-4. セーブデータを使用する場合の設定(Card2 のカードアプリ)

大項目	小項目	設定
Rom	SaveDataSize	「nMB」または「nGB」の形式で指定します。 1 MByte 単位でのみ指定が可能です。なお、BasicInfo / MediaSize で指定した値の半分が物理的な上限となります。

コード 4-4. RSF ファイルの記述例(Card2 のカードアプリでセーブデータを使用する場合)

```
Rom:
  SaveDataSize: 32MB
```

4.1.2.4. 体験版のセーブデータにアクセスする場合

体験版で作成したセーブデータにアクセスする必要があるアプリケーションは以下の項目を設定してください。

表 4-5. 体験版のセーブデータにアクセスする場合の設定

大項目	小項目	設定
AccessControlInfo	UseOtherVariationSaveData	「true」または「false」で指定します。 「true」の指定で、 nn::fs::MountDemoSaveData() による体験版のセーブデータのマウントが可能になります。 省略した場合は「false」を指定したことになります。

補足: AccessControlInfo の AccessibleSaveDataIds を指定した場合、そこにアプリケーションのユニーク ID が設定に含まれていなくてもアプリケーションのセーブデータと拡張セーブデータへのアクセスが可能になるように、UseOtherVariationSaveData を「true」に指定した際の解釈が拡張されます。

4.1.2.5. ほかのアプリケーションのセーブデータにアクセスする場合

ほかのアプリケーションのセーブデータにアクセスする必要があるアプリケーションは以下の項目を設定してください。

表 4-6. ほかのアプリケーションのセーブデータにアクセスする場合の設定

大項目	小項目	設定
AccessControlInfo	AccessibleSaveDataIds	アクセスしたいセーブデータのユニーク ID と拡張セーブデータ ID を、ユニーク ID の一覧で指定します。最大 6 個の指定が可能です。 アクセス可能になるのは、指定された ID と一致するユニーク ID のセーブデータ(体験版含む)と、指定された ID と一致する拡張セーブデータ ID の拡張セーブデータです。 この設定を使用した場合は UseOtherVariationSaveData の解釈が拡張されます。詳細については、「4.1.2.4. 体験版のセーブデータにアクセスする場合」を参照してください。
	OtherUserSaveDataId1 OtherUserSaveDataId2 OtherUserSaveDataId3	アクセスしたいセーブデータのユニーク ID を最大 3 個指定することができます。アプリケーションのユニーク ID と同じユニーク ID のセーブデータならば、この設定に含まれていなくても UseOtherVariationSaveData に「true」を指定することで使用可能になります。 AccessibleSaveDataIds が指定されている場合は、これらの設定を使用することができません。

4.1.3. 拡張セーブデータを使用する場合

アプリケーションで拡張セーブデータを使用する場合は、以下の項目を設定してください。

表 4-7. 拡張セーブデータを使用する場合の設定

大項目	小項目	設定
AccessControlInfo	AccessibleSaveDataIds	アクセスしたいセーブデータのユニーク ID と拡張セーブデータ ID を、ユニーク ID の一覧で指定します。最大 6 個の指定が可能です。 アクセス可能になるのは、指定された ID と一致するユニーク ID のセーブデータ(体験版含む)と、指定された ID と一致する拡張セーブデータ ID の拡張セーブデータです。 アプリケーションのユニーク ID をこの設定に含めた場合、UseOtherVariationSaveData の指定にかかわらず、拡張セーブデータへのアクセスが可能になります。 この設定を使用した場合は UseOtherVariationSaveData の解釈が拡張されます。詳細については、「4.1.2.4. 体験版のセーブデータにアクセスする場合」を参照してください。
AccessControlInfo	UseExtSaveData	「true」を指定します。 AccessibleSaveDataIds が指定されている場合は使用することができません。
	ExtSaveDataNumber	通常はアプリケーションのユニーク ID を指定します。 拡張セーブデータを共有する場合は共有の ID (共有するタイトルのユニーク ID)を指定します。 例えば、あるタイトルの続編が前作の拡張セーブデータを共有する場合は、前作のユニーク ID を指定します。 AccessibleSaveDataIds が指定されている場合は使用することができません。

コード 4-5. RSF ファイルの記述例(拡張セーブデータ)

```
AccessControlInfo:
  AccessibleSaveDataIds:
    - 0xFF3FF
```

以下のようなケースでは、どのアプリケーションのユニーク ID とも重複しない新しい拡張セーブデータ番号が必要になることがあります。

- 複数のアプリケーションがそれぞれ自分自身の拡張セーブデータを持ち、それとは別に共有するための拡張セーブデータを利用したい場合
- パッチを適用して、新たに拡張セーブデータ内のファイルを追加したいが、既存の拡張セーブデータに格納可能なファイル、ディレクトリの最大数を超過してしまう場合
- そのほかの理由で、ほかのアプリケーションとは独立して 1 つのアプリケーションで複数の拡張セーブデータを利用したい場合

上記のような場合に拡張セーブデータ番号を新しく取得したい場合は弊社窓口までご相談ください。

補足: AccessControlInfo の AccessibleSaveDataIds が指定されている場合は、UseExtSaveData および ExtSaveDataNumber の指定を使用することができません。この場合、アクセスしたい拡張セーブデータの拡張セーブデータ ID を AccessibleSaveDataIds に含めるか、その拡張セーブデータ ID とアプリケーションのユニーク ID が同じならば UseOtherVariationSaveData に「true」を指定することで使用可能になります。

注意: ファイルシステムのアクセス権利属性 (AccessControlInfo の FileSystemAccess) に「Debug」を指定すると、SD カードへの直接アクセスが可能になります。それに付随して、製品版ではアクセスできないデータ (AccessibleSaveDataIds に指定されていない別アプリのセーブデータと拡張セーブデータ、OtherUserSaveDataID* に指定されていない別アプリのセーブデータ、ExtSaveDataNumber に指定されていない拡張セーブデータ) へのアクセスも可能となります。そのため、動作検証は「Debug」を指定していない状態で行うようにしてください。

4.1.4. マスター提出用 ROM 作成時の設定

マスター提出用に ROM を作成するときは、以下の項目を追加もしくは開発時に指定していた内容から変更しなければなりません。

表 4-8. マスター提出用 ROM 作成時の設定

大項目	小項目	設定
BasicInfo	Title	アプリケーションのタイトルを指定します。 OMakefile の TITLE を参照する場合は、「\$(TITLE)」を指定してください。
	ProductCode	弊社発行の製品コードまたは追加コンテンツコードを指定します。
	MediaSize	メディアサイズを指定します。 MediaType が Card1 の場合は、「128MB」、「256MB」、「512MB」、「1GB」、「2GB」、「4GB」のいずれかを指定します。 MediaType が Card2 の場合は、「512MB」、「1GB」、「2GB」のいずれかを指定します。 ダウンロードアプリの場合は、カードアプリにした場合に必要になる値を指定してください。

	Logo	ロゴデータの種類を指定します。 ● 任天堂タイトル:「Nintendo」 ● 任天堂タイトル(販売のみ):「Distributed」 ● ライセンシータイトル:「Licensed」
TitleInfo	UniqueId	弊社発行のユニーク ID を指定します。 試作用コードを使用していた場合は、AccessControlInfo の ExtSaveDataNumber の指定も確認してください。

コード 4-6. RSF ファイルの記述例(マスター ROM 提出時)

```
BasicInfo:
  Title:          $(TITLE)
  ProductCode:    "CTR-A-00001"
  MediaSize:      512MB
  Logo:           Licensed
TitleInfo:
  UniqueId:       0x41001
```

4.1.5. リマスターバージョンの設定

すでにリリースされているアプリケーションに修正を加えた場合、リマスターバージョンを更新しなければなりません。

リマスターバージョンを更新した場合、カードアプリはすでに販売されているソフトウェアをバージョンアップすることはできませんが、ダウンロードアプリはユーザーの手元で遊ばれているソフトウェアをバージョンアップすることができます。

表 4-9. リマスターバージョンの設定

大項目	小項目	項目の内容
SystemControlInfo	RemasterVersion	CXI のリマスターバージョンを指定します。

コード 4-7. RSF ファイルの記述例(リマスターバージョン)

```
SystemControlInfo:
  RemasterVersion: 1
```

ダウンロードアプリのバージョンには、この CXI のリマスターバージョンがそのまま使用されます。CXI のリマスターバージョンが 1 であれば、ダウンロードアプリのバージョンも 1 となります。ダウンロードアプリをリリースしたあと、CXI のバージョンアップを行うごとに 1 ずつ増加していくことになります。

バージョンアップが行われる場合はセーブデータが維持されるため、新しいバージョンのダウンロードアプリは過去にリリースされたすべてのバージョンのセーブデータを適切に扱えるよう実装する必要があります。セーブデータのフォーマットを恒久的に変更しないか、あるいは事前にバージョン番号を埋め込んでおきプログラムで変更に対応できるようにしてください。

なお、ダウンロードアプリのバージョンアップ前後で RSF の下記の項目を変更することは禁止されています。

表 4-10. ダウンロードアプリのバージョンアップ時に変更が禁止されている項目

大項目	小項目	項目の内容
BasicInfo	Title	アプリケーションのタイトル

	ProductCode	製品コード
TitleInfo	UniqueId	ユニーク ID
Rom	SaveDataSize	セーブデータ領域のサイズ

4.2. BSF ファイル

BSF ファイルは ctr_makebanner で ICN ファイル (アイコンデータ) と BNR ファイル (バナーデータ) を作成する際に必要となる設定ファイルです。

BSF ファイルには、アイコンデータとバナーデータの作成に使用する、アイコンファイル、モデルファイル、サウンドファイルを指定します。これらのファイルは事前に作成しておく必要があります。

LittleIconFile と BigIconFile に指定するアイコンファイルには ctr_TexturePackager で作成する ctpk ファイルを使用します。

ModelFile に指定するモデルファイルには ctr_BannerModelConverter で作成する cbmd ファイルを使用します。

SoundFile に指定するサウンドファイルには ctr_WaveConverter で作成する bcwav ファイルを使用します。

また、HOMEメニューなどで表示されるタイトル名や、カードリージョン等のアプリケーションの設定も行います。

詳しくは CTR-SDK のツール「ctr_makebanner」のリファレンスを参照してください。

補足: 子機プログラムには BNR ファイルが必要ありませんので、子機プログラム用にモデルファイルとサウンドファイルを用意する必要はありません。

コード 4-8. BSF ファイルの記述例

```
LittleIconFile: resources/little.ctpk
BigIconFile:    resources/big.ctpk
ModelFile:      resources/model.cbmd
SoundFile:      resources/sound.bcwav

Region:         Japan
Ulcd:           True
AutoSave:       False
SaveData:       True
ThumbnailFrame: 0

RatingRequired: True
CERO:           12

JPLongName:     テストアプリケーション
JPShortName:    テストアプリケーション
JPPublisher:    テスト

ENLongName:     TestApplication
ENShortName:    TestApplication
ENPublisher:    Test
```

4.3. 電子取扱説明書

電子取扱説明書のファイルは CTR-ManualTools を使用して作成します。アプリケーションに含める電子取扱説明書のファイル名は Manual.bcma でなければならないことに注意してください。

電子取扱説明書のバイナリファイル (*.bcma) のファイルサイズは、1 言語あたり最大 6656 KByte です。

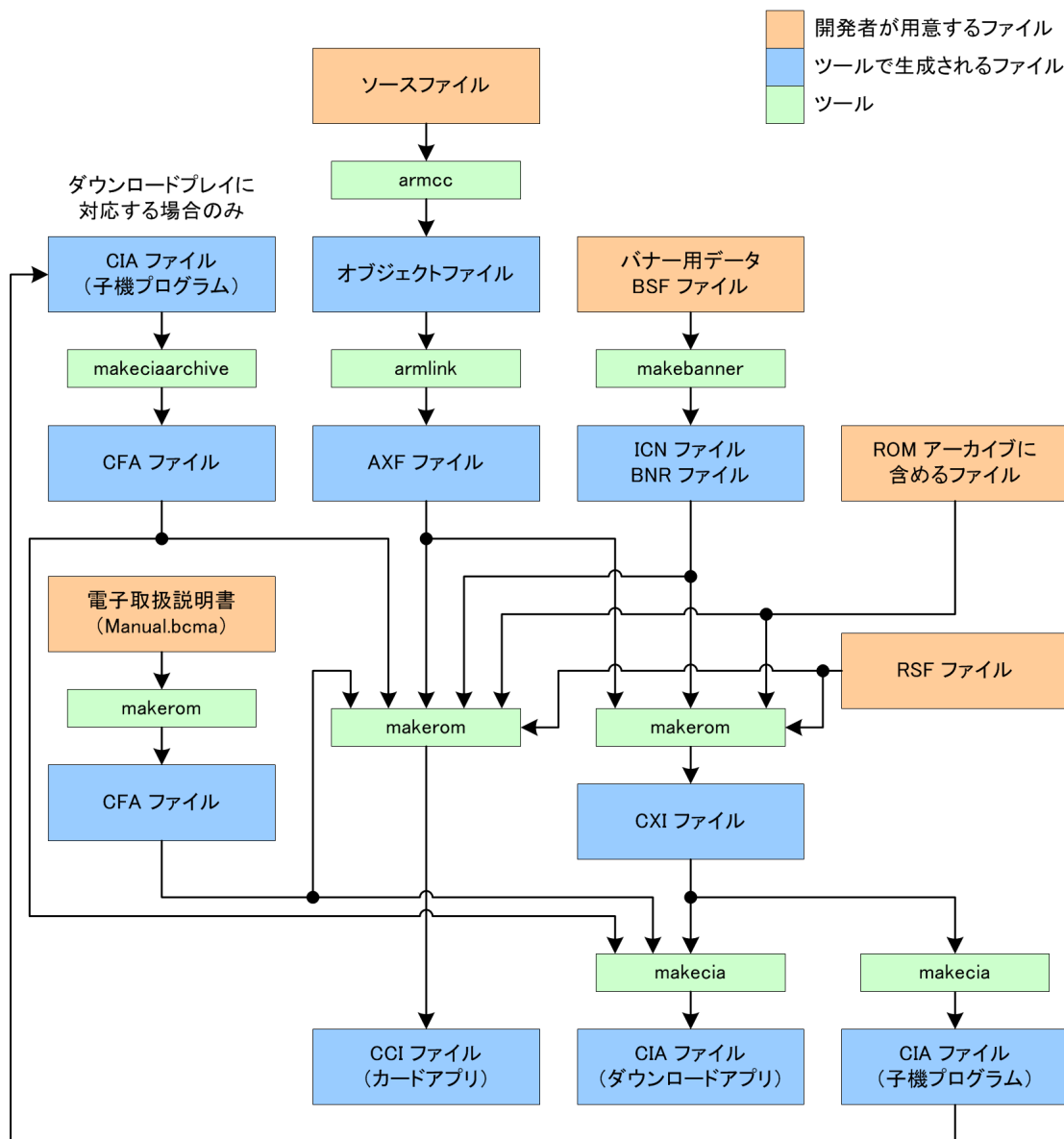
詳しくは「CTR-ManualTools」のドキュメントを参照してください。

5. ビルド

下図はアプリケーションをビルドするまでの概略をフロー図で示したものです。

ビルドに必要なファイルや処理についての詳細は、CTR-SDK の「ビルドシステム構築ガイド」を参照してください。

図 5-1. アプリケーションをビルドするまでの概略



5.1. CTR-SDK のビルドシステムでビルドする場合

CTR-SDK のビルドシステムを使用する場合、環境変数やビルドオプション、ビルド変数の設定を OMakefile に記述し、omake を実行するだけで簡単にアプリケーションの実行ファイルを作成することができます。作成するアプリケーションの種類をカードアプリからダウンロードアプリに変更する場合でも、CTR_APPTYPE の指定と、いくつかの設定を変更するだけで行うことができます。

OMakefile で CTR_BANNER_SPEC を指定した場合、CTR_BANNER_SPEC で指定された BSF ファイルの内容に基づいて、

ICN ファイルと BNR ファイルがビルド時に作成されます。このため、CTR_BANNER_SPEC を指定した場合は、CTR_BANNER、CTR_ICON を指定していたり、CTR_NO_BANNER、CTR_NO_BANNER_ICON の値を指定していたとしても、その設定は無視されます。

詳細については CTR-SDK の「ビルドルール」を参照してください。

5.1.1. カードアプリの作成

カードアプリをビルドする OMakefile を記述する上で、必須もしくは対応すべき内容は以下のとおりです。

表 5-1. カードアプリをビルドする OMakefile に記述する内容

設定	内容
CTR_APPTYPE	「CARD」
DESCRIPTOR	「\$(CTRSdk_ROOT)/resources/specfiles/Application.desc」(標準アプリ) 「\$(CTRSdk_ROOT)/resources/specfiles/ExtApplication.desc」(拡張アプリ)
ROM_SPEC_FILE	RSF ファイルを指定します。
CTR_BANNER	BNR ファイルを指定します。
CTR_ICON	ICN ファイルを指定します。
MANUAL_DIR	電子取扱説明書のファイルが格納されているディレクトリを指定します。
CHILD_APPS[]	ダウンロードプレイに対応する場合、配信する子機プログラムの CIA ファイル(複数指定可能)を指定します。

コード 5-1. OMakefile の記述例(カードアプリ)

```
SUPPORTED_TARGETS = CTR-TS.Process.MPCore.*

～ 中略(ヘッダファイル、ソースファイル、ライブラリ、シェーダの指定など) ～

TARGET_PROGRAM = MyApp
TITLE =          MyApp
CTR_APPTYPE =    CARD
DESCRIPTOR =     $(CTRSdk_ROOT)/resources/specfiles/Application.desc
ROM_SPEC_FILE =  MyApp.rsf
CTR_BANNER =     MyApp.bnr
CTR_ICON =       MyApp.icn
ROMFS_ROOT =     romfiles
MANUAL_DIR =     manual
CHILD_APPS[] =
    ../MyChildApp/images/$(BUILD_TARGET_DIR)/$(BUILD_TYPE_DIR)/MyChildApp.cia

include $(ROOT_OMAKE)/modulerules
build: $(DEFAULT_TARGETS)
```

表にない設定については、CTR-SDK の「ビルドルール」を参照してください。

5.1.2. ダウンロードアプリの作成

ダウンロードアプリをビルドする OMakefile を記述する上で、必須もしくは対応すべき内容は以下のとおりです。

表 5-2. ダウンロードアプリをビルドする OMakefile に記述する内容

設定	内容
CTR_APPTYPE	「SD」
DESCRIPTOR	「\$(CTRSDK_ROOT)/resources/specfiles/Application.desc」(標準アプリ・製品版) 「\$(CTRSDK_ROOT)/resources/specfiles/DemoVersion.desc」(標準アプリ・体験版) 「\$(CTRSDK_ROOT)/resources/specfiles/ExtApplication.desc」(拡張アプリ・製品版) 「\$(CTRSDK_ROOT)/resources/specfiles/ExtDemoVersion.desc」(拡張アプリ・体験版)
ROM_SPEC_FILE	RSF ファイルを指定します。
CTR_BANNER	BNR ファイルを指定します。
CTR_ICON	ICN ファイルを指定します。
MANUAL_DIR	電子取扱説明書のファイルが格納されているディレクトリを指定します。
MANUAL_DISCLOSURE	あらかじめダウンロードで電子取扱説明書を見せる場合に public を指定します。見せない場合、指定は不要です。
CHILD_APPS[]	ダウンロードプレイに対応する場合、配信する子機プログラムの CIA ファイル(複数指定可能)を指定します。

コード 5-2. OMakefile の記述例(ダウンロードアプリ)

```
SUPPORTED_TARGETS = CTR-TS.Process.MPCore.*

～ 中略(ヘッダファイル、ソースファイル、ライブラリ、シェーダの指定など) ～

TARGET_PROGRAM = MyDownloadApp
TITLE =          MyApp
CTR_APPTYPE =    SD
DESCRIPTOR =     $(CTRSDK_ROOT)/resources/specfiles/Application.desc
ROM_SPEC_FILE =  MyDownloadApp.rsf
CTR_BANNER =     MyDownloadApp.bnr
CTR_ICON =       MyDownloadApp.icn
ROMFS_ROOT =     romfiles
MANUAL_DIR =     manual
MANUAL_DISCLOSURE = public
CHILD_APPS[] =
    ../MyChildApp/images/$(BUILD_TARGET_DIR)/$(BUILD_TYPE_DIR)/MyChildApp.cia

include $(ROOT_OMAKE)/modulerrules
build: $(DEFAULT_TARGETS)
```

表にない設定については、CTR-SDK の「ビルドルール」を参照してください。

5.1.3. 子機プログラムの作成

子機プログラムをビルドする OMakefile を記述する上で、必須もしくは対応すべき内容は以下のとおりです。

表 5-3. 子機プログラムをビルドする OMakefile に記述する内容

設定	内容
CTR_APPTYPE	「NAND」
DESCRIPTOR	「\$(CTRSDK_ROOT)/resources/specfiles/DlpChild.desc」(標準アプリ) ※ 拡張アプリの子機プログラムは作成できません。
ROM_SPEC_FILE	RSF ファイルを指定します。
CTR_NO_BANNER	「true」
CTR_ICON	ICN ファイルを指定します。

コード 5-3. OMakefile の記述例(子機プログラム)

```
SUPPORTED_TARGETS = CTR-TS.Process.MPCore.*

～ 中略(ヘッダファイル、ソースファイル、ライブラリ、シェーダの指定など) ～

TARGET_PROGRAM = MyChildApp
TITLE =          ChildApp
CTR_APPTYPE =    NAND
DESCRIPTOR =     $(CTRSDK_ROOT)/resources/specfiles/DlpChild.desc
ROM_SPEC_FILE =  MyChildApp.rsfc
CTR_NO_BANNER =  true
CTR_ICON =       MyChildApp.icn
ROMFS_ROOT =     romfiles

include $(ROOT_OMAKE)/modulerrules
build: $(DEFAULT_TARGETS)
```

表にない設定については、CTR-SDK の「ビルドルール」を参照してください。

5.2. CTR-SDK のビルドシステム以外でビルドする場合

CTR-SDK で提供されているビルドシステムを使用せず、独自のビルドシステムを構築する場合にアプリケーションを作成するまでに必要な手順を説明します。

詳細については CTR-SDK の「ビルドシステム構築ガイド」を参照してください。

5.2.1. AXF ファイルの作成

armcc でコンパイルしたオブジェクトファイルやライブラリファイル等を armlink でリンクし、AXF ファイルを作成します。

コマンドの実行例:

```
armlink --datacompressor off --keep=nnMain \
--scatter=$CTRSDK_ROOT/resources/specfiles/linker/CTR.Process.MPCore.ldscript \
-o MyApp.axf MyApp.o MyAppLib.a ...
```

詳しくは ARMCC のドキュメント、CTR-SDK の「ビルドシステム構築ガイド」を参照してください。

5.2.2. ICN ファイルと BNR ファイルの作成

アプリケーションで使用するアイコンデータとバナーデータのファイルを作成します。

コマンドの実行例:

```
ctr_makebanner32.exe MyApp.bsf
```

BSF ファイル以外の引数を指定しなかった場合、ICN ファイルと BNR ファイルは BSF ファイルの拡張子を置き換えて作成されます。上記の実行例の場合は MyApp.icn と MyApp.bnr の 2 ファイルが作成されます。

詳しくは「4.2. BSF ファイル」、CTR-SDK のツール「ctr_makebanner」のリファレンスを参照してください。

5.2.3. 電子取扱説明書の CFA ファイルの作成

アプリケーションに付属させる電子取扱説明書の CFA ファイルを作成するには、ctr_makerom32.exe を以下のオプションで実行します。電子取扱説明書のファイル (Manual.bcma) は、事前に CTR-ManualTools で作成してください。

- -f data
- -rsf \$CTRSdk_ROOT/resources/specfiles/Manual.rsfs
- -DROMFS_ROOT=<電子取扱説明書のファイル (Manual.bcma) が格納されているディレクトリ>
- -DMANUAL_DISCLOSURE=<あらかじめダウンロードで電子取扱説明書を見せるなら true、見せないなら false>
- -o <作成する CFA ファイルの名前>

コマンドの実行例:

```
ctr_makerom32.exe -DROMFS_ROOT=manual_dir -DMANUAL_DISCLOSURE=true -f data \  
-rsf $CTRSdk_ROOT/resources/specfiles/Manual.rsfs -o Manual.cfa
```

詳しくは「4.3. 電子取扱説明書」、CTR-SDK のツール「ctr_makerom」のリファレンスを参照してください。

5.2.4. カードアプリの作成

カードアプリのデバッグやマスターとして提出する ROM には、カードアプリの最終 ROM イメージ形式である CCI ファイルを使用します。

5.2.4.1. CCI ファイルの作成

CCI ファイルの作成には、以下のファイルが必要とします。

- AXF ファイル
- DESC ファイル (CTR-SDK に付属のファイル)
- RSF ファイル
- ICN ファイル
- BNR ファイル
- 電子取扱説明書の CFA ファイル (電子取扱説明書に対応する場合)
- 子機プログラムの CFA ファイル (ダウンロードプレイに対応する場合)

CCI ファイルを作成するには、以下のオプションを指定して ctr_makerom32.exe を実行します。

- -f card
- -rsf <RSF ファイル>
- -desc \$CTRSdk_ROOT/resources/specfiles/Application.desc

- 拡張アプリの場合は -desc \$CTRSDK_ROOT/resources/specfiles/ExtApplication.desc
- -icon <ICN ファイル>
- -banner <BNR ファイル>
- -content <CFA ファイル>:<インデックス>
 - CFA ファイルが電子取扱説明書のファイルの場合はインデックスに 1 を指定します。
 - CFA ファイルが子機プログラムのファイルの場合はインデックスに 2 を指定します。
- -o <作成する CCI ファイルの名前>

詳しくは CTR-SDK のツール「ctr_makerom」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makerom32.exe MyApp.axf -f card -rsf MyApp.rsfs \
-desc $CTRSDK_ROOT/resources/specfiles/Application.desc \
-icon MyApp.icn -banner MyApp.bnr \
-content Manual.cfa:1 -content MyChildApp.cfa:2 \
-o MyApp.cci
```

5.2.5. ダウンロードアプリの作成

ダウンロードアプリのデバッグやマスターとして提出する ROM には、ダウンロードアプリの最終 ROM イメージ形式である CIA ファイルを使用します。

CIA ファイルの作成には CXI ファイルが必要ですので、先に CXI ファイルを作成します。

5.2.5.1. CXI ファイルの作成

CXI ファイルの作成には、以下のファイルが必要とします。

- AXF ファイル
- DESC ファイル (CTR-SDK に付属のファイル)
- RSF ファイル
- ICN ファイル
- BNR ファイル

CXI ファイルを作成するには、以下のオプションを指定して ctr_makerom32.exe を実行します。

- -f exec
- -rsf <RSF ファイル>
- -desc \$CTRSDK_ROOT/resources/specfiles/Application.desc
 - 標準アプリの体験版は -desc \$CTRSDK_ROOT/resources/specfiles/DemoVersion.desc
 - 拡張アプリは -desc \$CTRSDK_ROOT/resources/specfiles/ExtApplication.desc
 - 拡張アプリの体験版は -desc \$CTRSDK_ROOT/resources/specfiles/ExtDemoVersion.desc
- -icon <ICN ファイル>
- -banner <BNR ファイル>
- -o <作成する CXI ファイルの名前>

詳しくは CTR-SDK のツール「ctr_makerom」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makerom32.exe MyDownloadApp.axf -f exec -rsf MyApp.rsfs \
-desc $CTRSDK_ROOT/resources/specfiles/Application.desc \
-icon MyApp.icn -banner MyApp.bnr \
```

```
-o MyDownloadApp.cxi
```

5.2.5.2. CIA ファイルの作成

CIA ファイルの作成には、以下のファイルを必要とします。

- CXI ファイル
- 電子取扱説明書の CFA ファイル
- 子機プログラムの CFA ファイル (ダウンロードプレイに対応する場合)

CIA ファイルを作成するには、以下のオプションを指定して `ctr_makecia32.exe` を実行します。

- `-i` <CXI ファイル>
- `-man` <電子取扱説明書の CFA ファイル>
- `-i` <子機プログラムの CFA ファイル>:2 (インデックスの指定が必要です)
- `-o` <作成する CIA ファイルの名前>

詳しくは CTR-SDK のツール「`ctr_makecia`」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makecia32.exe -i MyDownloadApp.cxi -man Manual.cfa -o MyDownloadApp.cia
```

5.2.6. 子機プログラムの作成

カードアプリやダウンロードアプリに含める子機プログラムは 1 つの CFA ファイルにまとめ、カードアプリやダウンロードアプリの作成時のオプションで指定します。

CFA ファイルの作成には CIA ファイル、CIA ファイルには CXI ファイルが必要ですので、CXI ファイル、CIA ファイル、CFA ファイルの順に作成します。

5.2.6.1. CXI ファイルの作成

CXI ファイルの作成には、以下のファイルを必要とします。バナーは表示されませんので、BNR ファイルは必要ありません。

- AXF ファイル
- DESC ファイル (CTR-SDK に付属のファイル)
- RSF ファイル
- ICN ファイル

CXI ファイルを作成するには、以下のオプションを指定して `ctr_makerom32.exe` を実行します。

- `-f` `exec`
- `-rsf` <RSF ファイル>
- `-desc` `$CTRSdk_ROOT/resources/specfiles/DlpChild.desc`
- `-icon` <ICN ファイル>
- `-o` <作成する CXI ファイルの名前>

詳しくは CTR-SDK のツール「`ctr_makerom`」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makerom32.exe MyChildApp.axf -f exec -rsf MyChildApp.rsf \  
-desc $CTRSdk_ROOT/resources/specfiles/DlpChild.desc \  
-icon MyChildApp.icn \  
-o MyChildApp.cxi
```

5.2.6.2. CIA ファイルの作成

CIA ファイルの作成には、以下のファイルを必要とします。

- CXI ファイル

CIA ファイルを作成するには、以下のオプションを指定して ctr_makecia32.exe を実行します。

- -i <CXI ファイル>
- -o <作成する CIA ファイルの名前>

詳しくは CTR-SDK のツール「ctr_makecia」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makecia32.exe -i MyChildApp.cxi -o MyChildApp.cia
```

5.2.6.3. CFA ファイルの作成

CFA ファイルの作成には、以下のファイルを必要とします。

- 配信する子機プログラムすべての CIA ファイル

CFA ファイルを作成するには、以下のオプションを指定して ctr_makeciaarchive32.exe を実行します。ダウンロードプレイで配信する子機プログラムは、1 つの CFA ファイルにまとめて格納されます。

- -cia <子機プログラムの CIA ファイルのリスト>
- -o <作成する CFA ファイルの名前>

詳しくは CTR-SDK のツール「ctr_makeciaarchive」のリファレンスを参照してください。

コマンドの実行例:

```
ctr_makeciaarchive32.exe -cia MyChildApp.cia -o MyChildApp.cfa
```

ここで作成した CFA ファイルは、カードアプリの作成時のオプション -content (インデックス 2) や、ダウンロードアプリ作成時のオプション -i (インデックス 2) の指定で使います。

6. デバッグ

デバッグには以下のハードウェアとデバッグソフトウェアの組み合わせを使用します。

表 6-1. デバッグに使用するハードウェアとデバッグソフトウェアの組み合わせ

ハードウェア	デバッグソフトウェア	3DS 専用アプリ	SNAKE 対応アプリ
PARTNER-CTR Debugger	PARTNER-CTR	○	×
IS-SNAKE-BOX	IS-CTR-DEBUGGER	○	○

6.1. 起動時に表示されるメニューによって異なるデバッグ方法

PARTNER-CTR Debugger や IS-SNAKE-BOX を起動したとき表示されるメニューは Config ツールの「Menu Setting」にある「Menu」で切り替えることができます。アプリケーションをデバッグする方法は表示されるメニューによって異なります。

6.1.1. 開発用メニューから起動する場合

通常のデバッグは CCI ファイルまたは CIA ファイルを PARTNER-CTR または IS-CTR-DEBUGGER に読み込ませることで行います。起動直後からデバッグを行うことができますので、この方法でのデバッグを推奨します。

ただし、CTR FLASH CARD に書き込まれたカードアプリは開発用メニューから起動しても、HOMEメニューから起動する場合と同じ方法でなければデバッグを行うことができません。DevMenu からダウンロードアプリを起動した場合も同じです。

6.1.2. HOMEメニューから起動する場合

HOMEメニューから起動した場合はデバッグソフトウェアを介していないため、そのままではデバッグすることができません。

以下の手順で HOMEメニューから起動したアプリケーションをデバッグすることができます。(カードアプリのほか、SD カードにインポートされたダウンロードアプリもデバッグすることができます)

【PARTNER-CTR を使用する場合】

補足: PARTNER-CTR のエミュレーションメモリにロードしたアプリケーションは、リセットすることで HOMEメニューに表示されます。

1. PARTNER-CTR を起動します。
2. HOMEメニューからアプリケーションを起動します。
3. PARTNER-CTR のコマンドウィンドウで ATTACHA コマンドを実行します。
 > ATTACHA
4. LS コマンドで、シンボルファイルをロードします。
 > LS MyApplicaton.axf

この方法では起動済みのアプリケーションのデバッグとなるため、起動直後の箇所のデバッグは行うことができません。

PARTNER-CTR のコマンドの詳細については、「PARTNER-CTR デバッグマニュアル」を参照してください。

【IS-CTR-DEBUGGER を使用する場合】

補足: IS-CTR-DEBUGGER に CCI ファイルまたは CIA ファイルを読み込みませることも HOME メニューに表示できます。

1. IS-CTR-DEBUGGER を起動します。
2. HOMEメニューからアプリケーションを起動します。
3. IS-CTR-DEBUGGER メニューの「デバッグ」→「実行中のプログラムにアタッチ」を選択します。
4. IS-CTR-DEBUGGER メニューの「ファイル」→「開く」を選択し、AXF ファイルを読み込みます。

この方法では起動済みのアプリケーションのデバッグとなるため、起動直後の箇所のデバッグは行うことができません。

6.2. cia ファイルをデバッガソフトウェアで実行する場合

cia ファイルを直接デバッガソフトウェアでロードしてデバッグする際に、C8804478 エラーが発生することがあります。これは、cia ファイルと同じディレクトリに cxi ファイルが存在せず、かつアプリケーションがデフォルトのユニーク ID を使用していなかった場合に発生します。

cia ファイルを直接デバッガソフトウェアから実行したい場合は、下記のいずれかの対応が必要です。

- アプリケーションをデフォルトのユニーク ID でビルドする
- cia と同じディレクトリに cxi ファイルを配置する

7. DLL(動的モジュール)の利用

DLL を利用することで、プログラムによるメモリの消費を抑えることができるほか、起動時には静的モジュールのみがロードされますので、起動までの時間(3DS ロゴの表示時間)を短縮することができます。

7.1. 実装

DLL を利用しない場合は 1 つのメインプログラムにすべてを実装しますが、利用した場合はプログラムの実装を複数の動的モジュールに分割し、必要なときに必要なモジュールをメモリにロードして実行することができます。ただし、小さな単位の動的モジュールに分割しすぎると、モジュールのロードに時間がかかったり、頻繁な入れ替えでメモリの管理が複雑になったりする可能性があります。

詳しくは、CTR-SDK の「DLL マニュアル」、「3DS プログラミングマニュアル – システム編」を参照してください。

7.2. ビルド前の準備

「4. ビルド前の準備」で行う手順と同じです。

7.3. ビルド

7.3.1. CTR-SDK のビルドシステムでビルドする場合

モジュールのソースの配置や OMakefile の記述方法に注意するだけで、容易に動的モジュールに対応したアプリケーションをビルドすることができます。ただし、アプリケーション(静的モジュール)と動的モジュールの親ディレクトリが同じになるように、モジュールごとにビルドディレクトリを作成し、ソースファイルと OMakefile を個別に配置する必要があることに注意してください。

動的モジュールのビルドのために、OMakefile には以下の設定が追加されています。

表 7-1. 動的モジュールのビルドで追加された設定

設定	内容
TARGET_MODULE	動的モジュールの名前を指定します。
MODULE_LIST	自分以外のモジュール(静的モジュール含む)をすべて指定します。
MODULE_EXPORT_TYPE	動的モジュールの公開種別を指定します。 <ul style="list-style-type: none">● SYMBOL: 名前(デフォルト)● INDEX: インデックス● OFFSET: オフセット
MODULE_ENABLE_OFFSET	デフォルトでは true に設定されていますので、すべてのモジュールを同時にビルドする必要があります。false に設定した場合はモジュールのビルドタイミングが自由になりますが、公開種別にオフセットを指定することができなくなります。
MODULE_ADD_DEBUG_INFO	作成される CRR ファイルに CRO ファイルのデバッグ情報を含めるかどうかを指定します。デフォルトは true に設定されていますので、デバッグでのソースデバッグが可能になります。false に設定した場合はソースデバッグができなくなりますが、CRR ファイルのサイズが小さくなります。

動的モジュールのビルドに、RSF ファイル、ICN ファイル、BNR ファイル、DESC ファイルの指定は不要です。また、

TARGET_PROGRAM ではなく、TARGET_MODULE に動的モジュールの名前を指定することに注意してください。ビルドの結果、CRO ファイルが作成されます。

コード 7-1. 動的モジュールの OMakefile の例

```
TARGET_MODULE = Module1
MODULE_LIST = ../App ../Module2
SOURCES = module.cpp

include $(ROOT_OMAKE)/modulerrules
build: $(DEFAULT_TARGETS)
```

静的モジュールのビルドは、通常のビルド設定に MODULE_LIST の指定を追加するだけです。ビルドの結果、CRS ファイルと CRR ファイルが作成されます。

コード 7-2. アプリケーションの OMakefile に追加する設定の例

```
MODULE_LIST = ../Module1 ../Module2
```

作成された CRO ファイル、CRS ファイル、CRR ファイルは ROM アーカイブに格納する必要があります。CRO ファイルと CRS ファイルには、配置や圧縮に制限がありません。CRR ファイルには、ルート直下に作成した「.crr」ディレクトリに無圧縮で配置し、そのディレクトリには CRR ファイル以外を格納しないという強い制限があります。

以下の例は、動的モジュールを利用したアプリケーションのビルドで、各ファイルの格納を OMake で自動化する方法です。

コード 7-3. 動的モジュールのファイルの格納を自動化する OMakefile の例

```
MODULE_NAMES      = Module1 Module2
MODULES_ROOT      = ../

MODULE_CRO        = $(addprefix $(ROMFS_ROOT)/, $(addsuffix .cro,
$(MODULE_NAMES)))
STATIC_CRG        = $(ROMFS_ROOT)/static.crg
STATIC_CRR        = $(ROMFS_ROOT)/.crr/static.crr
ROMFS_DEPENDENCIES = $(MODULE_CRO) `$(STATIC_CRG) `$(STATIC_CRR)

foreach(TARGET, $(BUILDER.getTarget $(SUPPORTED_TARGETS)))
    foreach(module, $(MODULE_NAMES))
        src = $(file $(TARGET.getImageDirectoryOf
$(MODULES_ROOT)$(module))/$(module).cro)
        dst = $(addprefix $(ROMFS_ROOT)/, $(addsuffix .cro, $(module)))

        $(dst) : $(src)
            cp $< $@

        $(STATIC_CRG): $(TARGET.getImageDirectory false)/$(TARGET_PROGRAM).crg
            cp $< $@

        $(STATIC_CRR): $(TARGET.getImageDirectory false)/$(TARGET_PROGRAM).crr
            mkdir -p $(dirname $@)
            cp $< $@
```

詳細については CTR-SDK の「ビルドシステムマニュアル(DLL 編)」を参照してください。

7.3.2. CTR-SDK のビルドシステム以外でビルドする場合

CTR-SDK のビルドシステムを使用せずに動的モジュールをビルドするには、部分リンクやシンボルの参照解決など、特別な手順をビルドシステムに追加する必要があります。

詳細については CTR-SDK の「ビルドシステム構築ガイド (DLL 編)」を参照してください。

7.4. デバッグ

CRR ファイルに CRO ファイルのデバッグ情報が含まれるように設定してビルドした場合は、デバッガによるソースデバッグが可能になります。

CRR ファイルに CRO ファイルのデバッグ情報を含める方法については、「7.3.1. CTR-SDK のビルドシステムでビルドする場合」、CTR-SDK の「ビルドシステム構築ガイド (DLL 編)」を参照してください。

8. ダウンロード販売可能なカードアプリの開発手順

任天堂は CTR の本体システム開発当初から、カードアプリとして開発されたソフトがそのままニンテンドーeショップでも販売できるようにすることを目指してきました。このため、CTR-SDK はカードアプリとダウンロードアプリでコードを書き分ける必要性を極力排除してデザインされており、ガイドラインにおいてもアプリケーションへの要請事項は可能な限りカードアプリとダウンロードアプリで共通になっています。

この章では、同一のアプリケーションをカードアプリとダウンロードアプリの両方で販売できるようにする（併売対応する）ための手順を、新規に開発するカードアプリを併売対応するケースと、発売済みのカードアプリをダウンロードアプリとして販売するケースに分けて説明します。また、アプリケーションの修正が必要になった場合の手順も説明します。

注意： ダウンロードアプリ (cia) としてマスター提出した場合は、後からカードアプリとして販売を依頼することが出来ません。併売をご検討の場合はカードアプリ (cci) としてマスター提出するようにお願いします。

8.1. 新規に開発するカードアプリを併売対応するケース

ダウンロード版との併売に対応したカードアプリの新規開発は、基本的に通常のカードアプリを開発する場合と同じ手順で行います。ここでは、開発手順の中で留意すべき情報を説明します。

8.1.1. 企画段階

同一タイトルのカードアプリとダウンロードアプリは「販路の異なる同一の商品」という扱いです。**カードアプリとダウンロードアプリで挙動が異なることがないようにしてください。**

補足： カードアプリとダウンロードアプリで挙動に違いを出したい場合は、弊社窓口までご相談ください。
また、許可を得て挙動に違いをつける場合は、マスター提出時のチェックシートの「弊社承認仕様」に挙動の違いを具体的に記載する必要があります。

8.1.2. 取得する製品コードとユニーク ID

併売対応のアプリケーションでは、カードアプリ用の製品コードとユニーク ID をカードアプリとダウンロードアプリの両方で使用します。**パッケージ販売用とダウンロード販売用の 2 種類の製品コードおよびユニーク ID を取得しないように注意してください。**

例：カードアプリ用の製品コード「CTR-P-AAAJ(JPN)」を取得した場合は、ダウンロードアプリの製品コードも「CTR-P-AAAJ」としてください。ダウンロードアプリを意味する「CTR-N-AAAJ」は使用しません。

なお、製品コードに限り、実際の商品を管理する場合などに、ダウンロード版の製品コードを読み替えて（上記の例の場合は「CTR-N-AAAJ」と読み替えて）運用しますが、ソフト制作とマスター提出に関しては、「同じ製品コードとユニーク ID」と考えていただいて問題ありません。

8.1.3. 電子取扱説明書

必ず、カードアプリとダウンロード共通の電子取扱説明書を添付してください。

電子取扱説明書の作成には、「電子説明書制作マニュアル Part2: テンプレート編集」の最新版を使用してください。

8.1.4. セーブデータ、追加データ

1 台の本体で同一タイトルのパッケージ版とダウンロード版をプレイした場合、セーブデータについてはそれぞれ個別のデータ(パッケージ版はゲームカード内のバックアップメモリ、ダウンロード版は SD カード)にアクセスしますが、追加データ(拡張セーブデータ)については、双方が共通(同一)の追加データにアクセスすることになります。この状況にアプリケーションがどのように対応すべきかは、以下のように考えてください。

上述の状況は、同一タイトルの複数のゲームカードを 1 台の本体でプレイした場合にも発生します。ダウンロード販売に対応するかどうかに関わらず、カードアプリはこの状況で致命的な不具合が発生しないよう対応しているはずですので、これができていれば、ダウンロード販売のために追加で対応すべきことはありません。

「パッケージ版を所持していたユーザーが後日ダウンロード版を購入した際に、セーブデータを移行できる」といった特別な対応をアプリケーションで行う必要はありませんが、もしパッケージ版同士でデータ移行する機能などを実装している場合は、同等のサービスがパッケージ版とダウンロード版の間でも享受できることを確認してください。

8.1.5. セーブデータの巻き戻しについて

カードアプリの場合はユーザーの操作でセーブデータを以前の状態に巻き戻すということではできませんが、ダウンロードアプリの場合は PC を用いれば簡単に実現できてしまいます。セーブデータを保存したいタイミングで SD カードの内容すべてを PC にバックアップし、ゲームを進めたあとに PC にバックアップしたデータを SD カードに書き戻すことで、巻き戻しを実現できます。

`SetSaveDataSecureValue()` と `VerifySaveDataSecureValue()` (または `VerifySaveDataSecureValueLoosely()`) を用いることで、アプリケーションはセーブデータが巻き戻されたことを検出できます。カードアプリから呼ばれた場合、`SetSaveDataSecureValue()` は何も行わず、`VerifySaveDataSecureValue()` と `VerifySaveDataSecureValueLoosely()` は常に `true` を返します。つまり、カードアプリとして動作する場合は何も起こりませんので、これらの関数はダウンロード販売に対応したカードアプリでも使用することができます。

ゲームの企画上、セーブデータの巻き戻しがゲームバランス等に致命的な影響を与える場合は、これらの関数の使用をご検討ください。ただし、「巻き戻しを検出した際はセーブデータ全体を使えないようにする」というように、ユーザーに対して致命的なペナルティを設ける場合は以下のケースも想定されますので、ユーザーへの伝え方とサポート対応も十分に考慮の上、アプリケーションの仕様を決定してください。

- 巻き戻しに対してペナルティがあることを知らずに、データを PC にバックアップし、SD カードにある最新の `SetSaveDataSecureValue()` で記録した値に対応したデータを削除してしまった場合
ほかのゲームソフト(巻き戻し対策をしていないダウンロードアプリ)ではこのような操作を行っても問題が生じないため、やってはいけないことと認識されないまま、データを失ってしまう危険性があります。
- 本体修理で、SD カードのセーブデータはそのまま使えるよう救済できたが、本体保存メモリ内のデータは救済できなかった場合
`SetSaveDataSecureValue()` で記録した値が失われ、セーブデータ内の値との整合性が取れなくなります。このような状況では、`VerifySaveDataSecureValue()` を用いていた場合はペナルティが発動してしまいますが、`VerifySaveDataSecureValueLoosely()` を用いていた場合は `true` を返しますので、修理前に SD カードに入っていたセーブデータを使用することができます。ただし、修理前の SD カードに入っていたセーブデータが巻き戻されたデータである可能性はありますので、企画の面からこれが許容できるかを判断してください。

8.1.6. ゲーム内の表記、表示

ゲーム中のテキストメッセージや権利表記などは、ユーザーがカードアプリまたはダウンロードアプリのどちらで遊んでいても問題のないものにしてください。

以下は、カードアプリとダウンロードアプリの違いを考慮すべきゲーム内メッセージの一例です。

- 記録媒体の違い

カードアプリでは、ゲームカードにソフトとセーブデータが保存されています。

ダウンロードアプリでは、SD カードにソフトとセーブデータが保存されています。

ゲーム内メッセージの良い例と悪い例:

×:「セーブ中です。ゲームカードを抜かないでね。」

○:「セーブ中です。ゲームカード/SDカードを抜かないでね。」

○:「セーブ中です。カードを抜かないでね。」

- 同梱印刷物の有無

カードアプリには操作説明シートなどの印刷物が付属していますが、ダウンロードアプリにはこれらの印刷物が付属していません。

ゲーム内メッセージの良い例・悪い例:

×:「操作方法は付属の操作説明シートを見てね。」

○:「操作方法は電子説明書を見てね。」

○:「操作方法はゲーム内のヘルプを見てね。」

なお、ダウンロードアプリの場合とカードアプリの場合とを区別してユーザーに情報を伝えたい場合は、

ダウンロードアプリ → ダウンロード版

カードアプリ → パッケージ版

のように、ユーザー向け用語を使用してください。

8.1.7. ダウンロードプレイ

3DS のダウンロードプレイでは、親機側がカードアプリの場合はダウンロードプレイのシーケンス内でローカル通信を経由した本体更新 (DUP) が行われますが、親機側がダウンロードアプリの場合は DUP ができない仕様となっています。そのため、子機側本体のファームウェアバージョンが要求されるバージョンよりも古い場合、事前に本体更新がなされている必要があります。

このことをユーザーに適切に伝えるために、以下のいずれかの方法を実施してください。

- ダウンロードプレイの親機となる場合、子機を受け付けている間またはそれ以前のタイミングで、以下のメッセージを表示する。
「通信相手側に『本体の更新が必要です』と表示された場合は、ダウンロードプレイで遊ぶことができません。
通信相手側は『キャンセル』を選び、『本体設定』から本体の更新をしてください。『OK』を選んででも本体は更新されません。」

上記メッセージが長すぎて表示できない場合は、最低限「システムバージョンが古い本体とはダウンロードプレイできない」旨を表示してください。

メッセージは、`nn::dlp::Server (WithName)::Initialize()` の引数 `pNotice` に `true` がセットされている場合のみ表示してください。引数 `pNotice` が `false` となるのは、自身がカードアプリで DUP が可能な場合か、将来のシステムアップデートでダウンロードアプリでも DUP が可能になった場合です。

- 電子取扱説明書で、子機の本体更新が必要になる場合があることを説明する。
「電子説明書制作マニュアル Part2: テンプレート編集」に参考文例が入っています。

8.1.8. メディアへのアクセス速度

一般的に、記録メディアのアクセス速度に依存した実装を行うべきではないことはガイドラインの「ファイルシステム」にも記載されていますが、開発中のカードアプリをダウンロードアプリとして動作させた際に、ROM やセーブデータへのアクセス速度に差があることに留意してください。

ROM データのリード速度にはゲームカードと SD カードとの間で顕著な差はありませんが、ゲームカードのバックアップメモリに比べ、ダウンロードアプリのセーブデータへのリードおよびライトは格段に速くなります。

補足： アクセス速度についての詳細な情報については、「3DS パフォーマンス TIPS」を参照してください。

ROM データへのアクセスと追加データ (SD カード) へのアクセスを同時に行っている場合、カードアプリではアクセス先がそれぞれ別のメディアになりますが、ダウンロードアプリではアクセス先が共に SD カードになるため、同じ処理にかかる時間などに違いが出る可能性がありますのでご注意ください。

ファイルアクセスの優先度を設定することができます（「3DS プログラミングマニュアル – システム編」の「アクセス優先度の設定」参照）。たとえば、SD カードにあるストリーミングデータに対して高い優先度を設定しておかないと、カードアプリでは問題ないがダウンロードアプリではストリーミングが滞るといったことが起こる可能性があります。

8.1.9. アプリバイナリの生成(ビルド)

マスター提出時に CCI ファイル (カードアプリ用バイナリ) のみを提出いただく形となりますので、これまでのカードアプリと同じようにビルドを行ってください。

ダウンロードアプリとしての動作確認は、CCI ファイルを CIA ファイルに変換したものを SD カードにインポートして行ってください。

補足： CCI ファイルを CIA ファイルに変換するには makecia で -cci オプションを使用してください。

変換された CIA ファイルが正しく CCI ファイルから生成されたものかを確認するには MasterEditor を使用します。具体的には、CCI ファイルと CIA ファイルの対応する CXI タブと CFA タブ (複数ある場合はすべて) にある、「ハッシュ値 (CRC32)」欄の「コンテンツ全体」の値がそれぞれ等しいかどうかで確認します。

8.1.10. デバッグ

カードアプリとダウンロードアプリの両方でデバッグを行ってください。今までのデバッグ量の倍のデバッグを行っていただく必要はなく、デバッグに使用する本体のうちの何割かでダウンロードアプリのデバッグを行うようにしてください。

1 つの本体でカードアプリとダウンロードアプリの両方が動作するケースも、カードアプリのデバッグで複数のカードが 1 つの本体で動作することを確認すると同様にデバッグしてください。

デバッグの際には、開発実機に同梱されている SD カードまたは任天堂純正の SD カードでの確認を必ず行ってください。

補足： ガイドラインの「ファイルシステム」内にある「各メディアへのアクセス速度」を参照してください。

8.1.10.1. セーブデータの流用

CTR-SDK に付属している SaveDataFiler を使用することで、カードアプリのセーブデータを SD カードに書き出し、それをダウンロード版で読み込むことができます。

8.1.11. マスター提出

マスターデータ提出の手順は通常のカードアプリと変わりません。カードアプリとダウンロードアプリで同一のマスターデータ (CCI ファイル) を提出してください。ダウンロード販売用に CIA ファイルを別途提出する必要もありませんし、パッケージ版とダウンロード版それぞれで提出パッケージを作成する必要もありません。

提出の際は、CTR MasterEditor を用いて、「アプリの用途」に「一般販売」を選択し、「カード販売する」と「ダウンロード販売する」の両方にチェックを入れて**ダウンロード販売対応である旨を明示してください**。なお、**併売可能なアプリケーションとして提出するが、直ちにカード生産を行わずにダウンロード販売を先行したい場合は、あらかじめ弊社窓口にご連絡ください**。

弊社の許可を得てカードアプリとダウンロードアプリで挙動に違いをつけた場合は、**提出するチェックシートの「弊社承認仕様」の欄に挙動の違いを具体的に記載してください**。

8.2. 発売済みのカードアプリをダウンロードアプリとして販売するケース

すでに発売されているカードアプリをダウンロード版との併売に対応させるには、多くの場合、電子取扱説明書を新規に追加するだけで済む可能性があります。ここでは、開発手順の中で留意すべき情報を説明します。

8.2.1. SDK バージョン

カードアプリとして提出していたマスター ROM をダウンロードソフトとして使用する場合、もしくは、変更が電子取扱説明書の追加または差し替えのみの場合は、現時点での「マスター提出可能な SDK バージョンについて」のルールは不問になります。つまり、SDK のバージョンは当時のままで問題ありません。

何らかの変更を行ったためにリビルドする必要がある場合は、弊社窓口までご相談ください。ただし、変更内容が以下の場合は、SDK バージョンは当時のままで問題ありません。

- 一部のメッセージ・表示の変更などの軽微な修正
- 既にリリースされているパッチと同内容のリマスター

なお、変更を加えてリビルドを行った場合は必ず、**ロットチェックに合格した前回バージョンとの違いを具体的に記載した変更履歴を提出してください**。

8.2.2. 電子取扱説明書

原則として、「**電子説明書制作マニュアル Part2: テンプレート編集**」の最新版に記載されている**即時対応が必要な項目に対応している必要があります**。必要に応じて説明書を新規作成または改訂し、「CTR ROM エディタ」を用いて、発売済みのバイナリに添付または差し替えを行ってください。**説明書の添付により ROM サイズがアプリケーションの使用可能な領域サイズを超過してしまう場合は、メディアサイズも合わせて変更してください**。

表 8-1. 電子取扱説明書の新規作成や改訂の要・不要

電子取扱説明書	必要な変更
なし	電子取扱説明書を新規に作成し、添付してください。
あり	移行期間を問わず、全体の差し替えは不要です。発売当時の該当するバージョン以降の、即時対応が必要な箇所のみ差し替えてください。

補足: 現在、「電子取扱説明書制作マニュアル」は「CTR 電子説明書テンプレート集」に同梱されており、名称が「電子説明書制作マニュアル」に変更されています。

8.2.3. セーブデータの互換性について

併売対応時にアプリケーションの修正を行う場合や、カードアプリとして販売されているバージョンが改訂により複数存在する場合は、以下の点に注意してください。

販売されたカードアプリのいずれかのバージョンと、配信されるダウンロードアプリの間でセーブデータに互換性がない場合、セーブデータ移動ツールで移動したセーブデータが使用できないという致命的な問題が生じてしまいます。これに該当する場合は必ず弊社窓口までご連絡ください。セーブデータ移動ツールによって当該ソフトのセーブデータが移動できないように、弊社にて対応します。

8.2.4. セーブデータの巻き戻しについて

「8.1.5. セーブデータの巻き戻しについて」で説明した通り、ダウンロードアプリでは、ユーザーが PC を用いて SD カードのデータを操作することで、カードアプリのときには発生しなかったセーブデータの巻き戻りを起こすことができます。セーブデータの巻き戻りが生じてゲームバランスが破綻しないか、企画の面から確認してください。

8.2.5. ゲーム内の表記、表示

ゲーム中のテキストメッセージや権利表記などが、ダウンロード販売する際に問題ないかを確認してください。

補足: アプリケーションに変更がない場合、表記・表示に問題がないかの確認を改めて弊社が行うことはありません。

8.2.6. ダウンロードプレイ

ダウンロードプレイの挙動は「8.1.7. ダウンロードプレイ」で説明しています。**ダウンロードプレイに対応したアプリが、8.1.7 項に記載されているメッセージ表示を行っていない場合は、電子取扱説明書への記載で対応してください。**

8.2.7. 動作確認

ダウンロードアプリとして SD カードから実行した場合でも、不具合などが起こらないことを確認してください。

「8.1.4. セーブデータ、追加データ」、「8.1.8. メディアへのアクセス速度」、「8.1.9. アプリバイナリの生成(ビルド)」に記載されている点にも留意してください。

8.2.8. マスター提出

「CTR マスターROM提出手順書」にのっとり、電子取扱説明書が添付された CCI ファイルを再提出してください。

提出の際は、CTR MasterEditor を用いて、「アプリの用途」に「一般販売」を選択し、「カード販売する」と「ダウンロード販売する」の両方にチェックを入れて**ダウンロード販売対応である旨を明示してください。**

必要な提出物と付与すべきバージョンは下表の通りです。

表 8-2. マスター提出時に必要な提出物と付与すべきバージョン

	提出物				バージョン	
	マスターデータ 提出確認書	マスター ROM	チェック シート	変更履歴	リマスター バージョン	提出 バージョン
マスター ROM データが 前回提出時と全く同一の 場合	必要	不要	不要 (*1)	不要	前回と同じ	前回と同じ
説明書の追加または差し 替えのみの場合	必要	必要	不要 (*1)	必要	前回と同じ	前回 + 1
アプリが変更されている 場合	必要	必要	必要	必要	前回 + 1	0

(*1) カードアプリとダウンロードアプリで挙動に違いをつけていて、前回の提出書類にその旨が記載されていない場合は、「弊社承認仕様」の欄に挙動の違いを具体的に記載してチェックシートを提出してください。また、以下に該当するタイトルで前回の提出書類にその旨が記載されていない場合は、チェックシートを再提出してください。

- セーブデータ巻き戻し防止支援関数を使用するタイトル
- ほかのアプリケーションのセーブデータにアクセスするタイトル

補足: 提出物の作成には、基本的に最新バージョンの MasterEditor およびチェックシートエディターを使用してください。

ただし、最新バージョンの MasterEditor を使用すると以下のことが原因でエラーが発生することがあります。このエラーは、修正によるリビルドを行っていないのであれば無視してもかまいません。

- 長いタイトル、短いタイトル、パブリッシャー名について、値が設定されている項目と設定されていない項目が混在している。

8.3. アプリケーションの修正が必要になった場合

パッケージ版とダウンロード版の併売に対応したアプリケーションに対して、修正を行った場合に必要な手順を説明します。

修正は、基本的にはパッチによる対応を推奨します。詳しくは「3DS オーバービュー」の「パッチとは」を参照してください。

注意: 擬似クライアントを利用するタイトルを修正する場合は、リマスターバージョンの異なる擬似クライアントが DLP サーバーに接続する場合について配慮する必要があります。

詳しくは「パッチマニュアル」の「疑似クライアント機能」を参照してください。

8.3.1. 配信開始後に修正が必要になった場合

原則として、パッチを作成して提出してください。改訂版は提出不要です。

カードの再生産予定があり、修正内容を適用した改訂版を生産したい場合に限り、改訂版を提出してください。このときマスター提出確認書の「アプリの用途」欄は「カード販売する」のみにチェックを入れてください。このマスター ROM はカード生産用のみ使用され、ニンテンドーeショップでは配信されません。

修正が必要なほとんどの場合をパッチで対応できますが、パッチで対応できないケースが生じた場合は、併売用として改訂版を提出してください。このときマスター提出確認書の「アプリの用途」欄は「カード販売する」と「ダウンロード販売する」の両

方にチェックを入れてください。

補足: 説明書の改訂やダウンロードプレイの子機プログラムの修正もパッチで対応できます。

8.3.2. 発売済み、かつパッチをリリース済みのカードアプリを配信する場合

パッチ未適用の内容(カードで販売されているものと同内容)にて、「8.2. 発売済みのカードアプリをダウンロードアプリとして販売するケース」に記載されている手順に沿って併売対応し、再提出いただいて問題ありません。ただし、ダウンロードアプリにパッチを適用した場合でも正しく動作することを確認してください。

改訂版を提出いただいても問題ありません。

更新履歴

Version 1.3 2015-11-05

変更

- 2.2. ダウンロードアプリ
 - ダウンロードアプリは後からカードアプリとして販売できないことを追記しました。
- 8. ダウンロード販売可能なカードアプリの開発手順
 - ダウンロードアプリは後からカードアプリとして販売できないことを追記しました。
- 8.3. アプリケーションの修正が必要になった場合
 - 疑似クライアントを利用するタイトルを修正する場合の注意について追記しました。

Version 1.2 2015-04-28

追加

- 2.2.2. あらかじめダウンロード

変更

- 1.2. 用語解説
 - 改訂版、パッチについて解説を追加しました。
- 4.1.3. 拡張セーブデータを使用する場合
 - AccessibleSaveDataIds の詳細仕様を修正しました。
- 5.1.2. ダウンロードアプリの作成
 - ビルドオプション「MANUAL_DISCLOSURE」を追記しました。
- 5.2.3. 電子取扱説明書の CFA ファイルの作成
 - オプション「-DMANUAL_DISCLOSURE」を追記しました。
- 6.1. 起動時に表示されるメニューによって異なるデバッグ方法
 - Config ツールのメニュー階層を現行のツールに合わせて修正しました。

Version 1.1 2015-01-15

変更

- 4.1.3. 拡張セーブデータを使用する場合
 - AccessibleSaveDataIds の誤記を修正しました。
- 4.1.4. マスター提出用 ROM 作成時の設定
 - RSF ファイルの MediaSize 項目を修正しました。

削除

- プラットフォームの表記について
 - プラットフォーム表記を Readme に転記したため、ページを削除しました。

Version 1.0 2014-09-04

追加/変更

- 初版