

DoJa-5.0 API i α ppli Development Kit ユーザーズガイド

平成 19 年 3 月 9 日
株式会社 N T T ドコモ

改版履歴

版数	項目	種別	内容	
1.00	-	新規	-	2006年 6月 FOMA 903i 向け 制定 <iα ppli Development Kit for DoJa-5.0(FOMA)>
1.01	-	変更	改版内容	FeliCa の仕様変更に伴う修正
			改版理由	JBlend 部分の差し替えに伴う修正
1.02	-	変更	改版内容	仕様変更・不具合対応
			改版理由	JBlend 部分の差し替えに伴う修正
	5.6	変更	改版理由	不具合対応
			改版内容	903i 向けサウンドライブラリ対応
	5.3.9	変更	改版理由	機能追加
			改版内容	FeliCa の仕様変更に伴う機能拡張
1.03	-	変更	改版理由	機能追加
			改版内容	JBlend 部分の差し替えに伴う修正
	15.4.11	変更	改版理由	不具合対応
			改版内容	903i 向けサウンドライブラリ対応(Eclipse)
1.04	-	変更	改版理由	機能追加
			改版内容	JBlend 部分/903i 向けサウンドライブラリの差し替えに伴う修正
	表 5-2 10.15	変更	改版理由	不具合対応
			改版内容	ユーザーズガイドの不備を修正
2.00	-	追加	-	2006年 12月 FOMA 903i TV向け 制定
3.00	-	追加	-	2007年 3月 FOMA 904i 向け 制定

目次

第1章. はじめに	9
1.1. iappli Development Kit for DoJa-5.0(FOMA) の動作環境	9
1.1.1. iappliTool について	9
1.2. エミュレータの概要	10
1.2.1. エミュレータの利点	10
1.2.2. 統合開発環境の利用	10
1.2.3. エミュレーションの範囲	11
1.3. エミュレータによる i アプリの開発	14
第2章. インストール	16
2.1. インストールする前に	16
2.2. インストール手順	17
2.3. アンインストール手順	25
第3章. エミュレータの起動	26
3.1. ビルドエラービュー	27
3.2. プロジェクト・エクスプローラ	28
第4章. i アプリ	30
4.1. i アプリの作成から起動まで	30
4.2. プロジェクトの新規作成	31
4.2.1. ソースファイルの作成	33
4.2.2. ソースコードの修正	37
4.2.3. ソースエディタの設定	39
4.3. プロジェクトのビルドと通常起動	41
4.3.1. ダウンロード用HTMLファイル	44
4.3.2. i モード携帯電話で i アプリ実行	45
4.4. プロジェクトの読み込み	46
4.5. プロジェクトの設定	49
4.6. ファイル配置カスタマイズ	51
4.7. アプリケーションディスクリプタファイル (ADF) の編集	52
4.7.1. FeliCaCPIDの設定	63
4.7.2. FeliCaフリー領域の関連付け	64
4.7.3. LastModified項目の表記方法	65
4.8. 待ち受け起動	66
4.8.1. 待ち受けアプリケーション	66
4.8.2. 待ち受け起動手順	68
4.9. 受信フォルダ起動	71

4.1 0. 送信フォルダ起動	72
4.1 1. 未送信フォルダ起動	73
4.1 2. 初回起動	74
4.1 3. タイマ起動	75
4.1 4. 外部機器起動	76
4.1 4.1. 外部機器起動手順	76
4.1 5. 即起動	78
4.1 5.1. 即起動手順	78
4.1 6. メーラ起動	79
4.1 6.1. メーラ起動手順	79
4.1 7. ブラウザ起動	84
4.1 7.1. ブラウザ起動手順	84
4.1 8. コードリーダー起動	89
4.1 8.1. コードリーダー起動手順	89
4.1 8.2. CodeReaderの実行手順	93
4.1 9. 位置情報起動	94
4.1 9.1. 位置情報起動手順	94
4.2 0. 位置情報画像起動	97
4.2 0.1. 位置情報画像起動手順	97
4.2 1. 位置情報電話帳起動	100
4.2 1.1. 位置情報電話帳起動手順	100
4.2 2. アドホック起動	103
4.2 2.1. アドホック起動手順	104
4.2 2.2. i アプリが起動中の場合	106
4.2 3. トルカ起動	107
4.2 3.1. トルカ起動手順	107
4.2 4. 削除時起動	111
4.2 4.1. 削除時起動手順	111
4.2 5. BML 起動	112
4.2 5.1. BML 起動手順	112
4.2 6. DTV 起動	116
4.2 6.1. DTV 起動手順	116
4.2 7. 各種起動方法による画面遷移	120
第5章. エミュレータの設定	122
5.1. ネットワークの設定	122
5.1.1. プロキシサーバの設定	124
5.2. トラストッド動作の設定	125
5.3. メールアドレスの設定	128

5.4. リモート制御の設定	130
5.5. リモート制御のパラメータ領域の削除	131
5.6. サウンド設定	132
5.7. ヒープサイズの設定	135
5.8. U A R T用COMポートの設定	136
5.9. Bluetooth設定	138
5.10. 仮想FeliCa設定	141
5.11. ハイパーマルチ機能設定	142
5.12. 電話帳の編集	144
5.13. グループの編集	148
5.14. ブックマーク一覧	150
5.15. スケジュール一覧	151
5.16. 送受信メールの編集	152
5.16.1. ショートメールの扱いについて	156
5.16.2. メールを送受信エミュレート機能	158
5.17. 位置情報編集	159
5.18. 測位データ編集	162
5.19. コンパス情報編集	167
5.20. マイプロフィール編集	168
5.21. トルカデータ編集	169
5.22. デコメールテンプレート編集	172
5.23. 視聴/録画予約一覧	174
5.24. 加速度センサデータ編集	176
5.25. 動作認識データ編集	179
5.26. ビルドの設定	183
5.26.1. スクリプトの実行	183
5.26.2. ターゲットプロファイルの設定	186
5.27. A D F・トラステッド動作設定チェック処理の設定	187
5.28. ショートカットキーの設定	189
5.29. ヘルプの設定	192
5.30. エミュレータ性能設定	194
5.31. エミュレーション端末ウィンドウの設定	196
5.32. タイトル設定	198
5.33. スクリーンショット保存の設定	199

5.3 4. コンパイラ (javac) の選択	200
5.3 5. 起動時にプロジェクトを読み込む	201
5.3 6. .jam に関連付ける	202
5.3 7. コンソールクリア	202
5.3 8. エミュレーション端末の選択	203
5.3 9. エミュレーション端末の操作	204
5.3 9.1. 端末状態ウィンドウの操作	209
5.3 9.2. オプションキーウィンドウの操作	210
5.3 9.3. 3Dサウンド表示	212
第6章. ツール	214
6.1. スクラッチパッド管理ツール	214
6.1.1. スクラッチパッドファイル	214
6.1.2. スクラッチパッド管理ツールの起動	214
6.1.3. スクラッチパッドで J a r ファイルを扱う場合の注意事項	218
6.2. スクラッチパッドを全て初期化	218
6.3. 仮想FeliCaデータファイルビューワ	219
6.3.1. 仮想FeliCaデータファイル	219
6.3.2. 仮想FeliCaデータファイルビューワの起動	219
6.3.3. 仮想FeliCaデータファイルビューワの表示色	222
6.3.4. ブロックの構成	222
6.4. Javadoc 作成ツール	223
6.5. ADF チェックツール	226
6.6. スクリーンショット	228
6.7. i アプリライブラリ	229
6.7.1. i アプリライブラリの作成	229
6.7.2. i アプリライブラリの利用	236
6.8. CPBindingAuth署名付与ツール	238
6.8.1. ドコモが発行したユーザ証明書を使って署名する方法	238
6.8.2. ユーザが発行した証明書を使って署名する方法	239
第7章. エミュレータの終了	240
第8章. デバッグトレース	240
第9章. i アプリ連携	242
第10章. ファイルの記述方法	243
10.1. メーラ起動ファイルの記述方法	245
10.2. バウンダリデータファイル	245
10.3. メールアドレスファイル	246
10.4. ブラウザ起動HTMLファイルの記述方法	246

1 0.5. 着信履歴ファイル	247
1 0.6. メールファイル	247
1 0.7. 電話帳ファイル	248
1 0.8. 電話帳グループファイル	248
1 0.9. マルチメディア領域	249
1 0.1 0. ブックマーク登録ファイル	249
1 0.1 1. スケジュール登録ファイル	249
1 0.1 2. コード認識結果ファイル	250
1 0.1 3. L E D表示点灯パターン登録ファイル	252
1 0.1 4. ストレージデバイス ファイルパス	252
1 0.1 5. トルカファイル	252
1 0.1 6. ソーステンプレートファイル	254
1 0.1 7. 位置情報ファイル	254
1 0.1 8. 位置情報画像登録ファイル	254
1 0.1 9. 測位データ設定ファイル	255
1 0.2 0. スクリーンショットファイル	255
1 0.2 1. イルミネーションデータファイル	255
1 0.2 2. デコメールテンプレートファイル	256
1 0.2 3. コンパス設定ファイル	256
1 0.2 4. サウンド設定ファイル	257
1 0.2 5. マイプロフィール設定ファイル	257
1 0.2 6. 仮想FeliCaアドホック通信設定ファイル	258
1 0.2 7. Bluetooth設定ファイル	258
1 0.2 8. BML連携起動パラメータファイル	259
1 0.2 9. D T V連携起動パラメータファイル	259
1 0.3 0. 視聴/録画予約情報ファイル	260
1 0.3 1. 「アプリケーション名.properties」 ファイル	261
1 0.3 2. 仮想FeliCaデータファイル	262
1 0.3 3. スクラッチパッドデータファイル	263
1 0.3 4. ハイパーマルチデータファイル	263
1 0.3 5. 加速度センサデータファイル	263
1 0.3 6. 動作認識データファイル	264

1 0.3 7. CSV入出力ファイル	264
1 0.3 8. ファイルの記述具体例	265
第 1 1 章. 証明書の管理	266
第 1 2 章. OBEXの利用	268
第 1 3 章. 再配布不可識別子	268
第 1 4 章. Sun Java Studio Enterprise / NetBeans IDE の利用	269
1 4.1. Sun Java Studio Enterprise および NetBeans IDEについて	269
1 4.2. 「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iappli」 のインストール	269
1 4.2.1. インストール	269
1 4.2.2. IDEにモジュールを組み込む	270
1 4.3. 新規アプリケーションの作成	274
1 4.3.1. プロジェクトの作成	274
1 4.3.2. ソースファイルの作成	276
1 4.3.3. アプリケーションディスクリプタファイル (ADF) の設定	278
1 4.3.4. トラステッド動作の設定	279
1 4.3.5. アプリケーションの構築と実行	280
1 4.4. エミュレータの設定	282
1 4.4.1. デバイスの設定	282
1 4.4.2. ネットワークの設定	282
1 4.4.3. IDE からモジュールを削除する	283
第 1 5 章. Eclipse の利用	284
1 5.1. Eclipseについて	284
1 5.2. 「Eclipseプラグイン for iappli」 のインストール	285
1 5.2.1. 始める前に	285
1 5.2.2. インストール	285
1 5.3. 新規アプリケーションの作成	286
1 5.3.1. DoJaを利用する前に	286
1 5.3.2. プロジェクトメニューのカスタマイズ	287
1 5.3.3. プロジェクトの作成	288
1 5.3.4. ソースファイルの作成	291
1 5.3.5. プロジェクト設定からのソースファイルの作成	294
1 5.3.6. アプリケーションディスクリプタファイル (ADF) 設定	295
1 5.3.6.1. FeliCaCPIDの設定	296
1 5.3.6.2. FeliCaフリー領域の関連付け	296
1 5.3.7. トラステッド動作の設定	297
1 5.3.8. アプリケーションの構築と実行	298
1 5.3.9. プロジェクトの設定	300
1 5.3.10. リモート制御の設定	301
1 5.3.11. ファイル配置カスタマイズ	302
1 5.3.12. i アプリライブラリの作成	303
1 5.3.13. i アプリライブラリの利用	304

1 5.3.1 4. デバッグの実行	307
1 5.3.1 5. i アプリからD T V起動を行う場合の注意事項	314
1 5.4. エミュレータの設定	315
1 5.4.1. インストール先の設定	315
1 5.4.2. ヒープサイズの設定	315
1 5.4.3. ターゲットプロファイルの変更	316
1 5.4.4. ネットワークの設定	319
1 5.4.5. メールアドレスの設定	320
1 5.4.6. 電話帳の編集	321
1 5.4.7. グループの編集	324
1 5.4.8. 送受信メールの編集	325
1 5.4.9. スケジュール一覧	327
1 5.4.1 0. ブックマーク一覧	328
1 5.4.1 1. サウンド設定	329
1 5.4.1 2. 位置情報編集	330
1 5.4.1 3. 測位データ編集	332
1 5.4.1 4. コンパス情報編集	334
1 5.4.1 5. マイプロフィール情報編集	335
1 5.4.1 6. ToruCaデータ編集	336
1 5.4.1 7. デコメールテンプレート編集	338
1 5.4.1 8. Bluetooth設定	340
1 5.4.1 9. 仮想FeliCa設定	341
1 5.4.2 0. 視聴/録画予約一覧	342
1 5.4.2 1. ハイパーマルチ機能設定	343
1 5.4.2 2. 加速度センサデータ編集	344
1 5.4.2 3. 動作認識データ編集	346
1 5.5. 仮想FeliCaデータファイルビューワ	348
1 5.5.1. 仮想FeliCaデータファイルビューワの起動	348
1 5.5.2. 仮想FeliCaデータファイルビューワの表示色	350
1 5.5.3. ブロックの構成	350
1 5.6. スクラッチパッド	351
1 5.6.1. スクラッチパッド管理ツール	351
1 5.6.2. スクラッチパッド管理ツールの起動	351
1 5.6.3. スクラッチパッドを全て初期化する	353
1 5.6.4. スクラッチパッドでJ a r ファイルを扱う場合の注意事項	353
付録.1 エミュレーション端末の設定	354
1. サウンドライブラリの設定	354
用語の説明	355

本製品または文書は著作権法により保護されており、その使用、複製、再頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。NTT ドコモ（その他に許諾者がある場合は当該許諾者も含めて）の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。フォントを含む第三者のソフトウェアは、著作権法により保護されており、その提供者からライセンスを受けているものです。

Sun、Sun Microsystems、Sun Java Studio Enterprise、NetBeans、Java、J2SE SDK、JDK、J2ME および J2SE は、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。サンのロゴマークは、米国 Sun Microsystems, Inc. の登録商標です。Windows、DirectX は、米国およびその他の国における米国 Microsoft Corporation の登録商標です。

FeliCa は、ソニー株式会社が開発した非接触 IC カードの技術方式です。FeliCa は、ソニー株式会社の登録商標です。

QR コードは、株式会社デンソーウェーブの登録商標です。

本製品は、株式会社エイチアイの Mascot Capsule Engine Micro3D Edition を搭載しています。Mascot Capsule は、株式会社エイチアイの日本国における登録商標です。

本製品は、株式会社アプリックスの JBlend TM[micro] を搭載しています。JBlend は、株式会社アプリックスの商標または登録商標です。

CPBindingAuth 署名付与ツールは、Aicrypto を搭載しています。
Aicrypto :

This product includes software developed by Akira Iwata
Laboratory, Nagoya Institute of Technology in Japan
(<http://mars.elcom.nitech.ac.jp/>).

「i モード」、「i アプリ／アイアプリ」、「i-α ppli」ロゴ、「DoJa」、「i メロディ／アイメロディ」、「FOMA」は NTT ドコモの商標または登録商標です。

その他、掲載されている会社名、製品名、サービス名は各社の商標または登録商標です。

文書は現状有姿で提供されており、その商品性、特定目的への適合性、第三者の権利の非侵害を含み、明示的または黙示のいかなる条件、表明および保証も行われず、また提供されません。但し、かかる保証の否定が、適用される法律の下で無効である場合は、この限りではありません。

Copyright 1997-2007 Aplix Corporation. All Rights Reserved.

Copyright 2000-2007 NTT DoCoMo, Inc. 2-11-1 Nagata-cho, Chiyoda-ku, Tokyo, 100-6150, Japan.

第1章. はじめに

本書では、「iα ppli Development Kit for DoJa-5.0(FOMA)」のインストール、設定、および使用方法について説明します。

ここで述べる「iα ppli Development Kit for DoJa-5.0(FOMA)」は、iモード携帯電話で動作する Java アプリケーション（i アプリ）の開発をサポートします。

本書を理解するためには、Java プログラミング、CLDC、および NTT ドコモが提供する DoJa-5.0 API、DoJa-5.0 オプション・拡張 API の知識が必要になります。

i アプリでできることについては、別途、i アプリコンテンツ開発ガイド、もしくは、API リファレンスを参照してください。

1.1. iα ppli Development Kit for DoJa-5.0(FOMA) の動作環境

「iα ppli Development Kit for DoJa-5.0(FOMA)」は、以下の環境での動作を保証します。

- Windows 2000 (SP4) / Windows XP SP2
- Java^(TM) 2 SDK, Standard Edition 1.4.2_04 (J2SE SDK)
- Java^(TM) 2 Runtime Environment, Standard Edition 1.4.2_04
- DirectX 8.1 以上（サウンドライブラリ利用時）

これらは保証できる範囲であり、動作する環境はこの限りではありません。

1.1.1. iα ppliTool について

「iα ppli Development Kit for DoJa-5.0(FOMA)」をインストールすると「iα ppliTool」という i アプリのエミュレータが実行できるようになります。

「iα ppliTool」（以降、エミュレータと呼びます）の起動方法とその詳細については、「第3章. エミュレータの起動」で説明します。

1.2. エミュレータの概要

エミュレータ (iαppliTool) は、i モード携帯電話でできる様々な動作を PC 上で実現することによって i アプリの開発をサポートします。このエミュレータを使用すれば、i モード携帯電話で動作する i アプリを PC 上で検証することができます。

1.2.1. エミュレータの利点

エミュレータ (iαppliTool) には、次の利点があります：

- 作成した i アプリの動作を検証できます。
- i アプリのライブラリを作成できます（「6.7. i アプリライブラリ」参照）。
- 作成した i アプリがそのまま i モード携帯電話で実行できます。
（Web 上にあれば、簡単にダウンロードして実行できます。）
- 様々な携帯電話の機能をエミュレーションできます。

1.2.2. 統合開発環境の利用

統合開発環境 (IDE) は、アプリケーションの作成に必要な 編集、コンパイル、実行、デバッグ の機能が統合されており、アプリケーションの生産性を向上させることができます。「iαppli Development Kit for DoJa-5.0(FOMA)」では、Sun Java Studio Enterprise 8、NetBeans IDE 4.1 および Eclipse3.0/3.1 で i アプリを開発するために必要な拡張モジュールを提供します。これらの統合開発環境で、i アプリの編集、コンパイル、ビルド、実行、デバッグまでを一貫して行うことができます。

詳細については「第 1 4 章. Sun Java Studio Enterprise / NetBeans IDE の利用」、「第 1 5 章. Eclipse の利用」を参照してください。

- Sun Java Studio Enterprise IDE : <http://jp.sun.com/products/software/tools/jsenterprise/>
- NetBeans IDE : <http://www.netbeans.org/>
(**Mobility Pack は必要ありません。**
付属のモジュールは、最新のNetBeans IDE 5.0 でも使えますが、
動作は保証されません。)
- Eclipse : <http://www.eclipse.org/downloads/>

1.2.3. エミュレーションの範囲

ここでは、エミュレータがエミュレーションできる i モード携帯電話機能の範囲について説明します。

i アプリの動作をテストするときに理解しておく内容ですので、まず i アプリを作成してみたい場合は、読み飛ばしても構いません。

1. アプリケーションマネージャの処理

エミュレータは、Java アプリケーションの実行環境を提供していますが、実際の i モード携帯電話で実行されるアプリケーションマネージャの動作をエミュレートする環境は提供していません。

アプリケーションマネージャに関する動作のうち、i モード携帯電話と異なる動作として次のものがあります：

- ADF のサイズに制限はありません。またスクラッチパッドを使用しないアプリケーションについてはアプリケーションのサイズにも制限はありません。
- アプリケーションのダウンロードや更新の機能はありません。ただし、起動種別設定機能を使用することにより IApplication#LAUNCHED_AFTER_DOWNLOAD を発生させるテストが可能です。
- ネットワーク接続可否の設定機能はありません。
- タイマ起動を設定する機能はありません。ただし、起動種別設定機能を使用することにより IApplication#LAUNCHED_FROM_TIMER を発生させるテストが可能です。
- 実際にメールやメッセージを送受信する事はできません。
- 実際に赤外線リモコンで赤外線を飛ばす事はできません。
- 実際に動画再生する事はできません。
- 実際にカメラ撮影、動画撮影をする事はできません。
- 実際にGPSで測位する事はできません。
- 実際にPKIのPKCS7データを作成、および、署名をする事はできません。
- 実際にBluetooth通信をする事はできません。
- 実際にDTVを視聴する事はできません。
- エミュレータはWindows上で動作するため、実際のiモード携帯電話と動作が異なる場合があります。

2. アプリケーションディスクリプタファイル (ADF)

エミュレータでは、アプリケーションディスクリプタファイルにおいて実際に処理される項目と処理されない項目があります。しかし、全ての項目において値の検証をしています。そのため、アプリケーションとして設定する必要のない項目については、値を設定しないことを推奨します。アプリケーションディスクリプタファイルについての詳細は「4.7. アプリケーションディスクリプタファイル (ADF) の編集」を参照してください。

3. 携帯電話固有の機能

エミュレータでは、携帯電話固有機能のうちバックライトやバイブレータの状態表示など一部の機能をエミュレートします。しかし、携帯電話固有機能の多くの機能はエミュレートされません。

携帯電話固有機能に関する動作のうち、i モード携帯電話と異なる動作として次のものがあります：

- バックライトやバイブレータの状態はエミュレータ上のアイコン領域に表示されます。
- 音量を設定する機能はありません。
- 電源の ON/OFF や自動電源 ON/OFF の機能はありません。
- i メロディ、着信メロディ、着信画像を設定する機能はありません。
- ダイヤルロック等、ロック機能や保護機能はありません。
- TV 電話機能はありません。

4. 待ち受けアプリケーション

エミュレータでは待ち受けアプリケーションのライフサイクルを実現していますが、待ち受けアプリケーションを登録するための設定機能は提供していません。

待ち受けアプリケーションに関する動作のうち、i モード携帯電話と異なる動作として次のものがあります：

- 待ち受け画面（待ち受け画像など）の設定/表示機能はありません。
- 待ち受けアプリケーションを i モード携帯電話に設定する機能はありません。
- 待ち受けアプリケーションが休眠状態の際に実行される省電力機能はエミュレートされません。

5. ネイティブアプリケーション連携

エミュレータでは、i モード携帯電話に搭載されたネイティブアプリケーションで、ブラウザ、メール、コードリーダー、音声通話の機能連携を提供しますが、外部機器との連携をエミュレートする環境は提供していません。

ネイティブアプリケーションとの連携に関する動作のうち、i モード携帯電話と異なる動作として次のものがあります：

- 外部機器からアプリケーションを起動する機能はありません。ただし、起動種別設定機能を使用することにより IApplication#LAUNCHED_FROM_EXT を発生させるテストが可能です。
- ブラウザから i アプリを起動する際、i モード通信可否の判定は行いません。また、以下の点に注意してください：

- ✧ <A>タグに ista 属性および ijam 属性があっても無視されます。また utn 属性が指定される場合も同様に無視されます。
- ✧ 1つの HTML 内には、<OBJECT>タグと<A>タグをこの順序で対にして記述してください。
- ✧ 対応関係にある<OBJECT>タグと<A>タグとの間には、他の<OBJECT>タグを書かないでください。
- ✧ 1つの HTML 内での各<A>タグにおける href 属性値は、異なる文字列として記述してください。
- ✧ <OBJECT>タグでは、data 属性だけが意味を持ち、id、type、declare 属性は意味を持ちません。
- ✧ 最低限、<OBJECT>タグ内においては、data 属性とその値（ADF の URL）、および、<A>タグ内においては、href 属性とその値を記述してください。

- メールから i アプリを起動する際、以下の点に注意してください：
 - ◇ "--B:A"以降の記述については、規定のフォーマットに従っていない場合は、全て無視します。
 - ◇ メール本文は、HTML フォーマットに対応しています。
- 6. HTTP を使用したアプリケーション
 - エミュレータでは、HTTP1.1 を用いており、pragma: no-cache は、cache-control: no-cache を設定して HTTP 要求を発行します。
 - Connection: keep-alive は、常に設定されて要求を発行します。
- 7. FeliCa を使用したアプリケーション
 - エミュレータでは FeliCa のフリー領域にアクセスする API を利用できる機能を提供します。
 - エミュレータでは FeliCa アドホック通信によるデータ転送をする API を利用できる機能を提供します。

1.3. エミュレータによるiアプリの開発

ここでは、エミュレータによる i アプリの開発サイクルについて説明します。

エミュレータを使用した場合、

i アプリの開発サイクルは、次のようなフェーズになります：

1. i アプリのプロジェクト作成
エミュレータで、i アプリのプロジェクトを作成します。
2. i アプリのプログラム (.java) を作成もしくは、修正
エミュレータでは、テンプレートを使って簡単なプログラムを作成できます。
3. i アプリで扱うリソース（画像や音楽、電話帳/メール等）を設定
4. アプリケーションディスクリプタファイル（ADF）を設定
5. プロジェクトのビルド
エミュレータでビルドを実行すると次の作業をまとめておこないます。
自動的に実行されるため、エミュレータを使って i アプリを作成する上では特に意識する必要はありません。
 - コンパイル
J2SE SDK コンパイラを使用して、ソースファイルをクラスファイルにコンパイルします。コンパイラは、CLDC や DoJa API に含まれていないパッケージやクラス、メソッドがアプリケーションで使用されていないか検証します。
 - バイトコードの検証
事前検証ツールを使用して、Java クラスの事前検証をすると共に、Java クラスの中で浮動小数点演算やファイナライズメソッドが使用されていないかどうかを検査します。
 - パッケージング
J2SE SDK の jar ツールを使用して、事前検証された全てのクラスファイルとリソースファイルを JAR 形式にパッケージングします。
 - アプリケーションディスクリプタファイル（ADF）の作成
アプリケーションディスクリプタファイルを作成します。これは、i アプリの情報を記述するテキストファイルです。
 - ダウンロード用 HTML の作成
作成した i アプリをダウンロードするためのもっともシンプルな HTML を作成します。
6. エミュレータで実行
エミュレータ上で i アプリを実行します。
もし、i アプリの動作に問題があれば、2. からやり直します。
7. i モード携帯電話で実行
作成した i アプリ一式を HTTP サーバ上に配置後、i モード携帯電話で i アプリをダウンロードして実行します。

上述の開発サイクルを図 1-1 に示します。

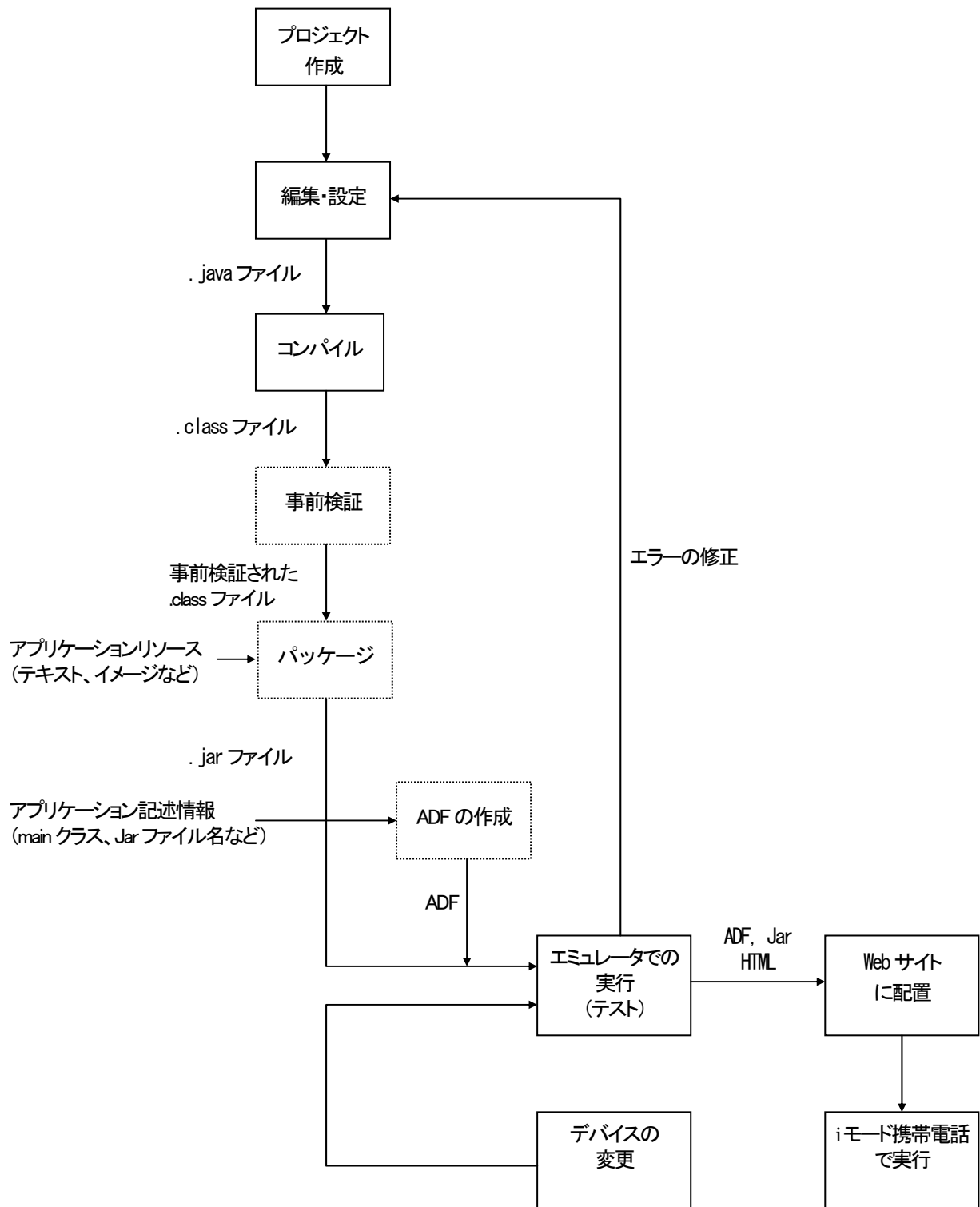


図 1-1 アプリケーション開発サイクル

第2章. インストール

この章では、エミュレータ（iαppliTool）のインストールとアンインストールの手順を説明します。

2.1. インストールする前に

「iαppli Development Kit for DoJa-5.0(FOMA)」をインストールする前に次のことを確認してください。

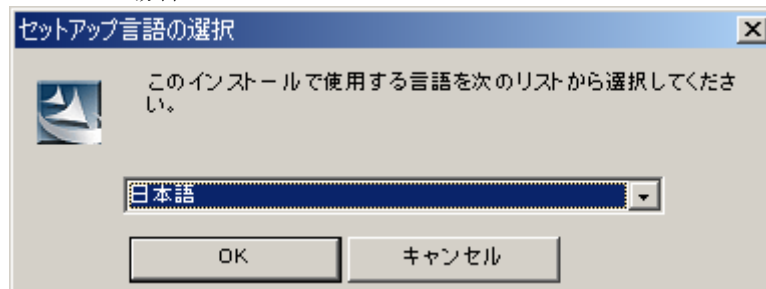
- ① 「iαppli Development Kit for DoJa-5.0(FOMA)」を動作させるには、**J2SE SDK (JDK)** がインストールされている必要があります。もし、J2SE SDK がインストールされていない場合は、Sun のサイトから J2SE SDK を入手してインストールしてください。
- ② エミュレータでは、より携帯端末に近いサウンドを再生するために **DirectX (Direct Sound) 8.1 以上** を使っています。ただし、エミュレータのサウンド設定が『ビジュアルモード』に設定されている場合は使われません。もし、DirectXが入っていない場合は、インストール時にサウンド設定が『ビジュアルモード』に変更されます。携帯端末に近い『通常再生モード』でサウンドを鳴動させたい場合は、事前に DirectX を入手してインストールしてください。
 - ・ DirectX : <http://www.microsoft.com/japan/windows/directx/>
- ③ 「iαppli Development Kit for DoJa-5.0(FOMA)」の以前のバージョンが既にインストールされている場合には、これをアンインストールしてください。アンインストールについては、「2.3. アンインストール手順」を参照してください。
- ④ Eclipse 統合開発環境を使う場合で、他のプロファイル（DoJa-4.0 や DoJa-3.5）と共存させたい時は、あらかじめ別の Eclipse 環境を用意しておいてください。
（例：C:\Eclipse_DoJa-5.0）
※ Sun Java Studio Enterprise 8 もしくは、NetBeans IDE では共存できません。
- ⑤ JDK は、JDK5.0 以降でも問題なく動作しますが、保証されるものは、JDK1.4.2 のみです。
（※ Sun Java Studio Enterprise 8 は、JDK5.0 以降です。）

2.2. インストール手順

「iα ppli Development Kit for DoJa-5.0(FOMA)」をインストールするには：

1. 「DISK1」ディレクトリの下にある「setup.exe」をダブルクリックします。
2. 最初にインストールする言語を聞いてきますので言語（日本語）を選択します。
英語版の Windows の場合、英語で表示されます。

日本語版 Windows の場合：



英語版 Windows の場合：

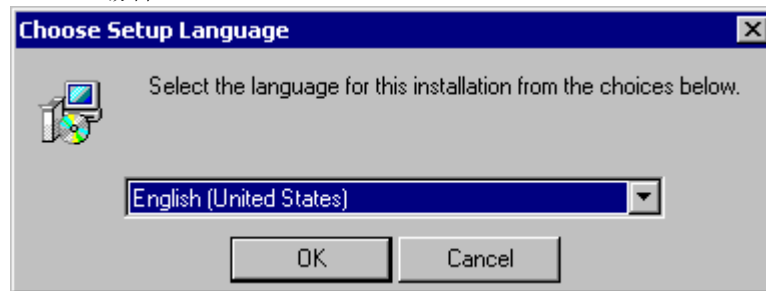


図 2-1 セットアップ言語の選択画面

3. ウィザードに従い「次へ」を押します。

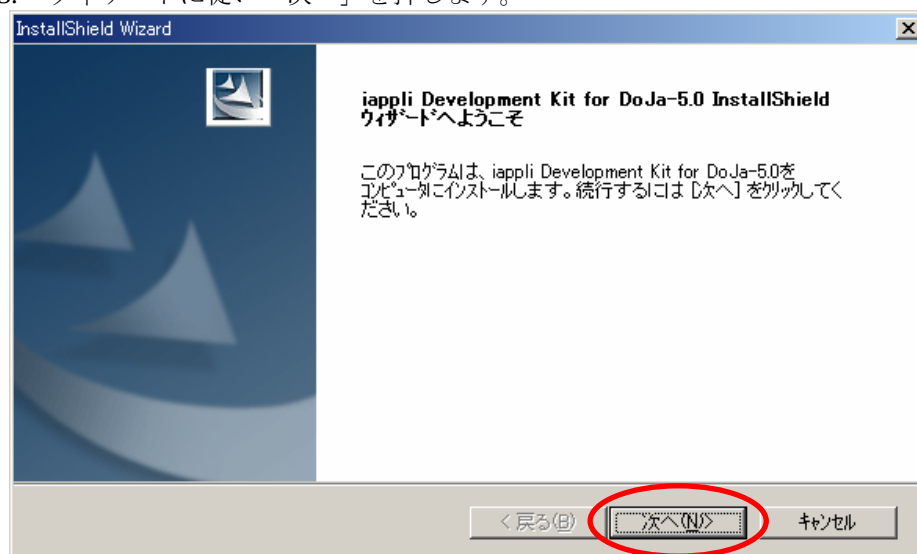


図 2-2 ようこそ画面

4. 「使用許諾契約」が表示されるので、よく読んでから「はい」を押します。

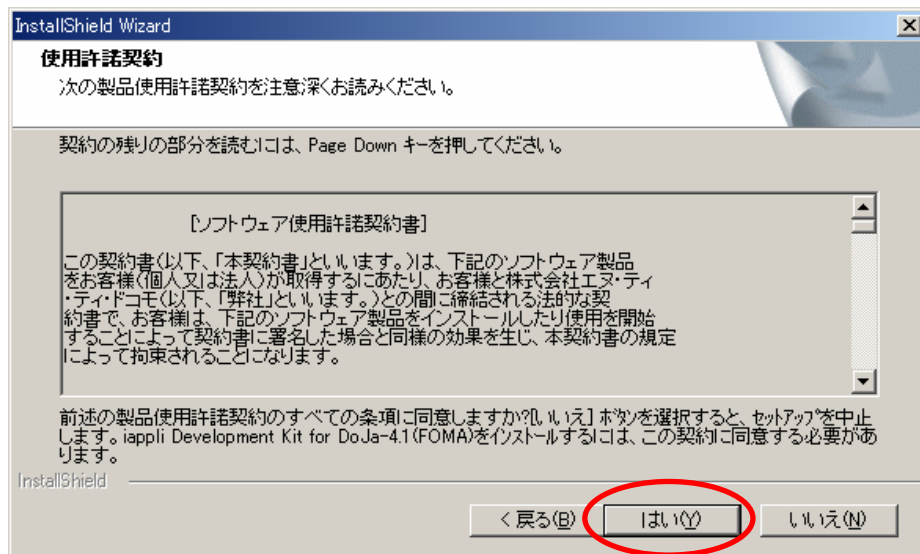


図 2-3 使用許諾契約画面

5. 「Binary Code License Agreement」が表示されるので、「はい」を押します。

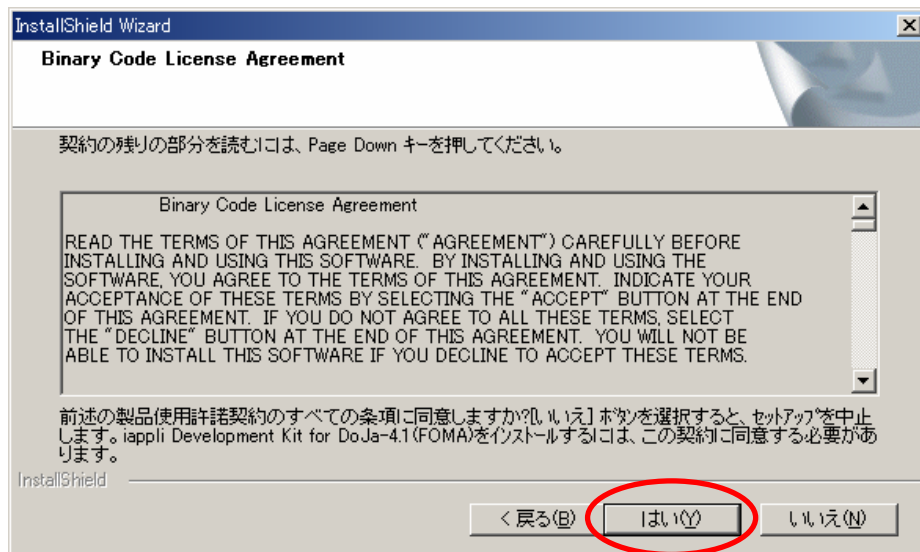


図 2-4 「Binary Code License Agreement」

6. インストール先のフォルダを指定^[1]して、「次へ」を押します。

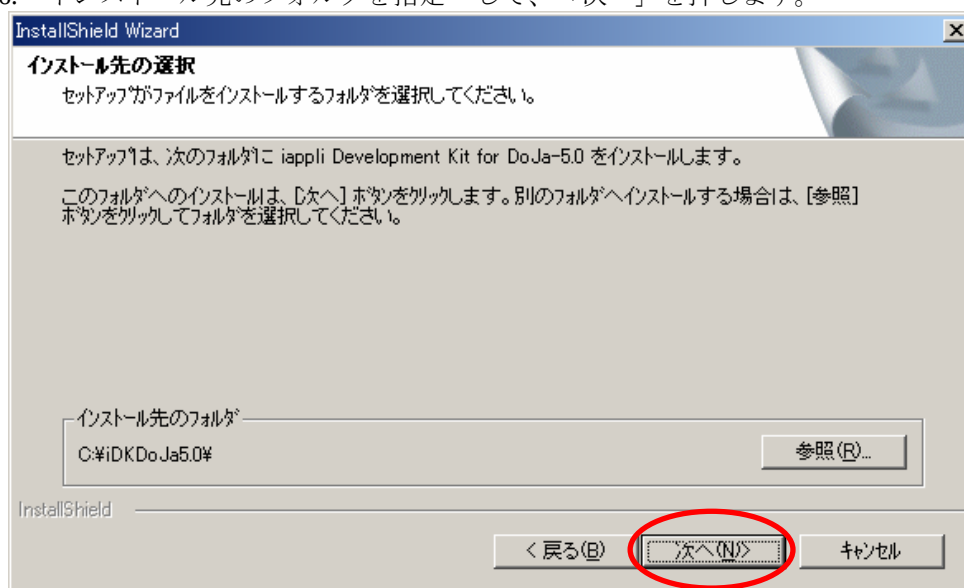


図 2-5 インストール先の選択画面

7. セットアップの種類を選択して、「次へ」を押します。

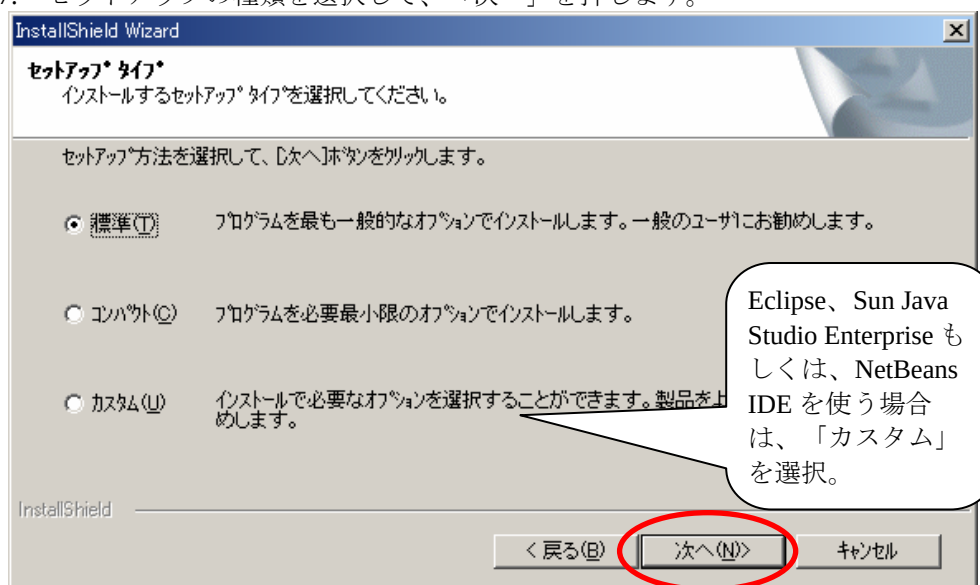


図 2-6 セットアップタイプの選択画面

- ・標準 : プログラムとサンプルアプリをインストールします。
- ・コンパクト : プログラムのみをインストールします。
- ・カスタム : 何をインストールするかを選択することができます。
もし、**Sun Java Studio Enterprise**、**NetBeans** や **Eclipse** を使ってアプリケーションを作成したい場合には、こちらを選択してください。

^[1] インストールのディレクトリを指定する際「スペース（空白）」及び「2バイト文字（全角文字）」を含まないようにしてください。これらの文字が含まれているとディレクトリを正しく扱うことができません。

8. 手順7で、「カスタム」以外を選択した場合は、手順11へ進んでください。
「カスタム」の場合、インストールしたい機能を選択して、「次へ」を押します。

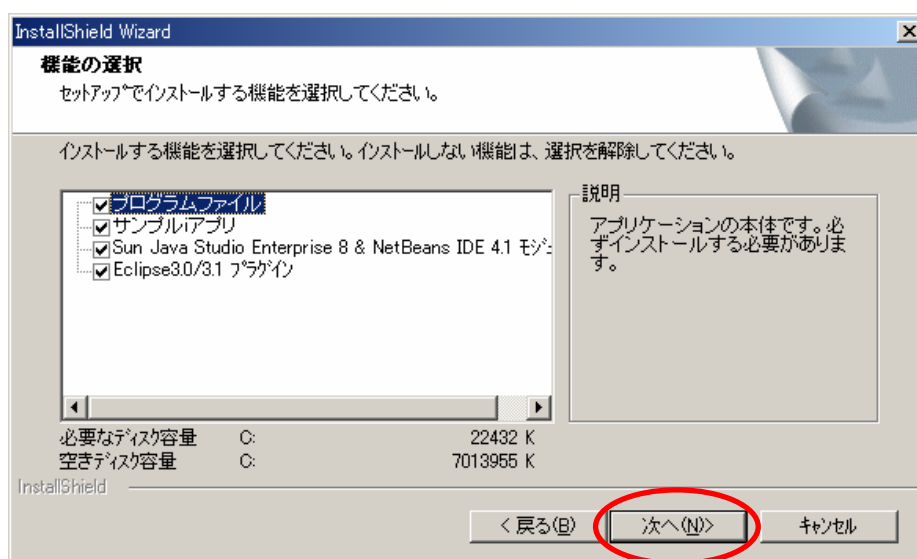


図 2-7 機能の選択画面

- 「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール」を選択すると、Sun Java Studio Enterprise 8 もしくは、NetBeans IDE 4.1 を使って i アプリを開発することができます。
ここでインストールされた **Sun Java Studio Enterprise 8 & NetBeans IDE 4.1** モジュールは、自動的に組み込まれませんので、Sun Java Studio Enterprise 8 もしくは、NetBeans IDE の機能「アップデートセンター」にて組み込んでください（「第14章. Sun Java Studio Enterprise / NetBeans IDE の利用」参照）。
（本モジュールは、最新のNetBeans IDE 5.0 でも利用可能ですが、動作は保証されません。）
- 「Eclipse3.0/3.1 プラグイン」を選択すると、Eclipse3.0 もしくは、Eclipse3.1 を使って i アプリを開発することができます。
ただし、あらかじめ Eclipse がインストールされている必要があります。
**注意：DoJa-5.0 以前のプラグインがインストールされている場合に、
DoJa-5.0 以前と共存させたい場合は、別の Eclipse 環境を用意してください。**

9. 手順8で「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール」を選択した場合、注意事項が表示されるので「OK」を押します。

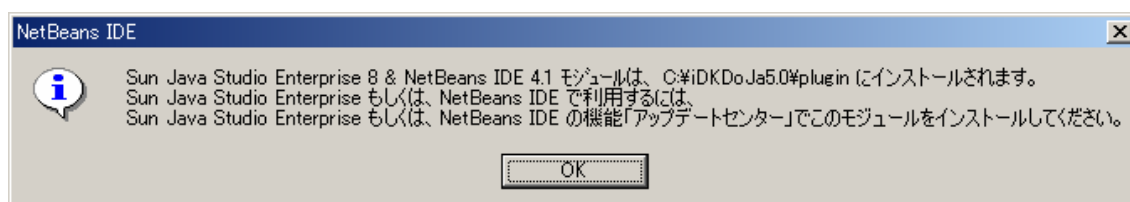


図 2-8 Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュールの注意事項

10. 手順 8 で「Eclipse3.0/3.1 プラグイン」を選択した場合、Eclipse のインストール先を聞いてきますので、インストール先を指定してください。

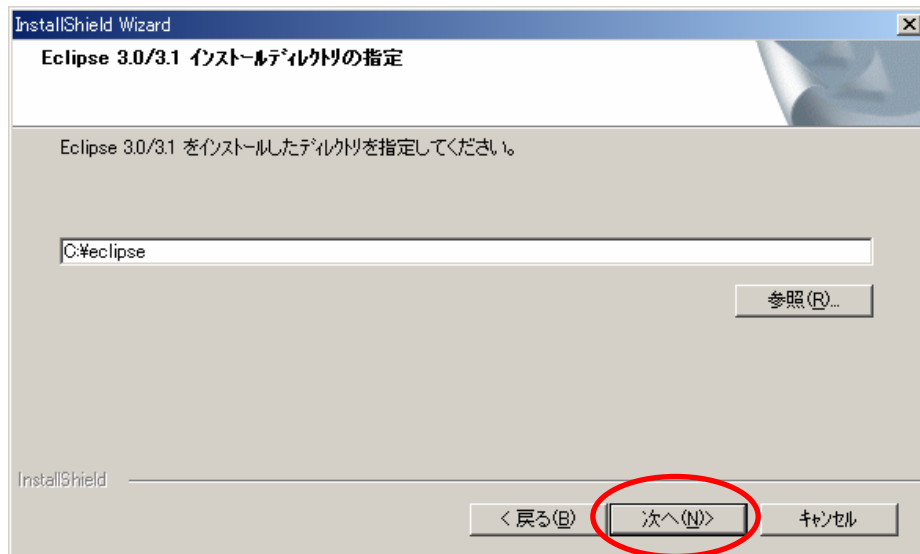


図 2-9 Eclipse プラグインインストール先設定画面

11. インストールされるファイルの確認画面が表示されるので「次へ」を押します。

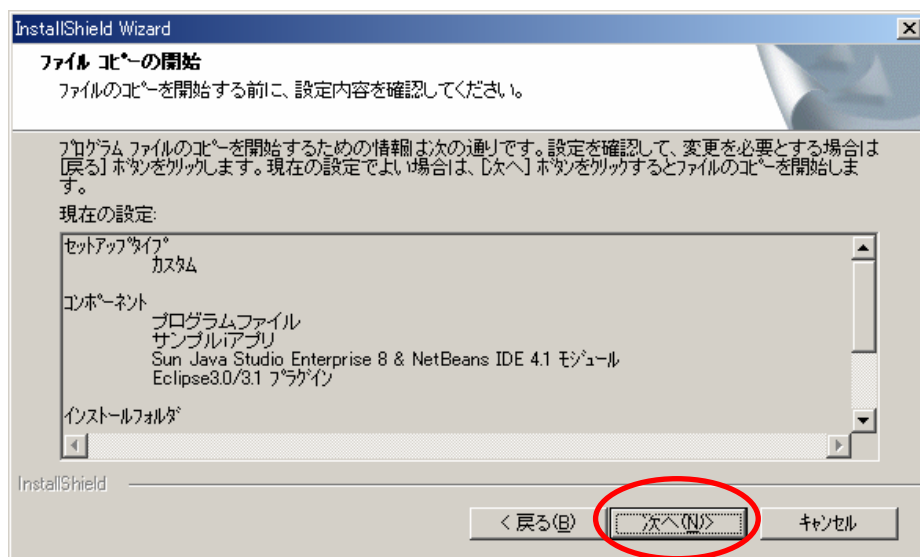


図 2-10 ファイルのコピー開始画面

12. Windows にプロキシサーバ情報が設定されている場合、エミュレータのネットワーク設定にプロキシサーバ情報を登録するかどうかの確認画面が表示されますので、プロキシサーバ情報を設定してもよろしければ、「はい」を押します。

Windows のプロキシサーバの情報を利用している場合：

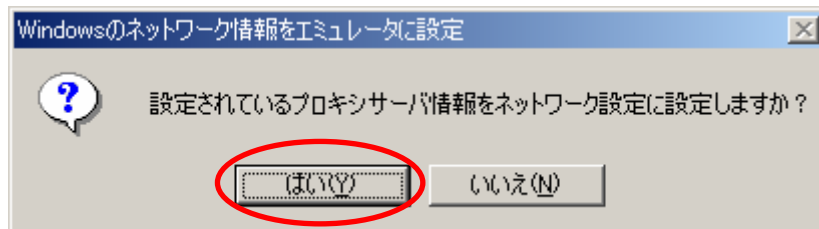


図 2-1 1 プロキシ情報の設定画面

Windows のプロキシサーバの情報を利用していない場合：

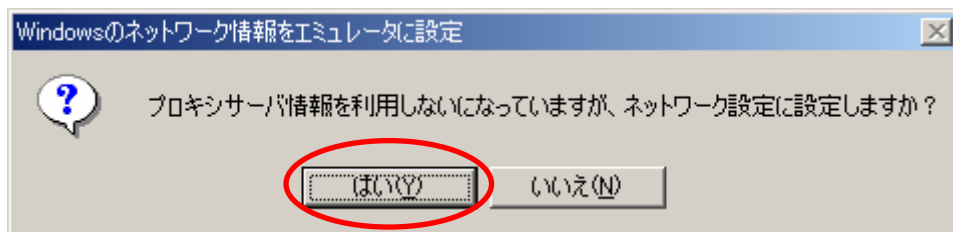


図 2-1 2 プロキシ情報の設定画面

13. i アプリ（.jam ファイル）をエミュレータに関連付けるかどうかを問い合わせるので、i アプリ（.jam ファイル）からエミュレータが起動できるようにしたい場合は、「はい」を押します。
関連付けをしない場合は、エミュレータを起動して i アプリを読み込むという手順になります。
拡張子の関連付けは、後からでも変更できます（「5.3 6. .jam に関連付ける」参照）。

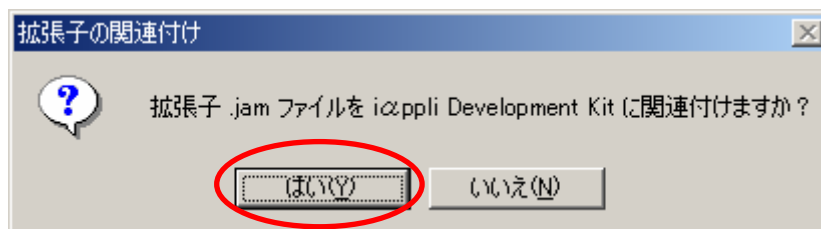


図 2-1 3 .jam ファイルの関連付け設定画面

14. インストールが実行され、完了画面が表示されたら「完了」を押します。

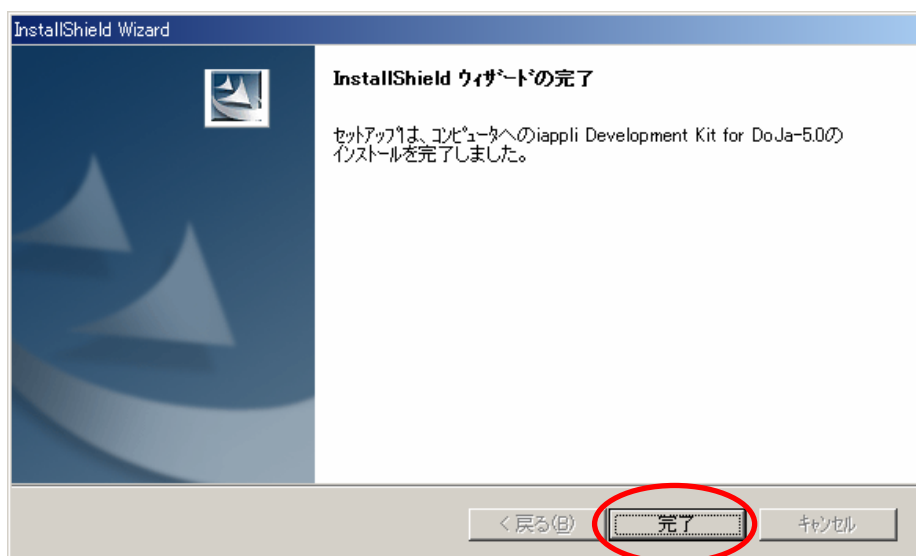


図 2-1 4 インストール完了画面

15. 日本語版Windowsで「iα ppli Development Kit for DoJa-5.0(FOMA)」をインストールした場合、i モード携帯電話と同じ「絵文字」を登録するための設定画面が表示されます。Windowsの外字領域（何も使われていないところ）に絵文字を設定してもよろしければ、そのまま「設定」ボタンを押します（図 2-1 5）。
もし、後から登録したい場合は、「iα ppli Development Kit for DoJa-5.0(FOMA)」をインストールしたディレクトリの「gaiji」ディレクトリにある「ChangeEUDC.exe」を実行してください。

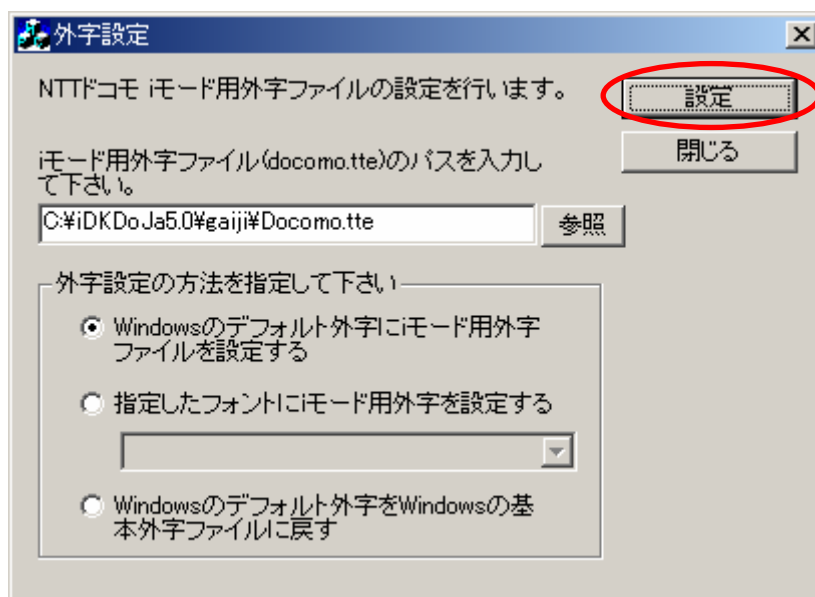


図 2-1 5 「ChangeEUDC.exe」の実行（外字設定画面）

16. 設定終了の確認画面が表示されたら、「OK」を押します（図 2-1 6）。

17. 外字設定画面の「閉じる」を押します（図 2-1 5）。



図 2-1 6 絵文字（外字）設定終了の確認

一度、インストールした絵文字（外字）をアンインストールするには、図 2-1 5 の「外字設定の方法」の一番下「Windowsのデフォルト外字をWindowsの基本外字ファイルに戻す」を選択して「設定」ボタンを押します。

18. i アプリオプションAPIと呼ばれる各メーカー機種固有のAPIを使う場合は、各メーカーのサイトからライブラリ（DLL）を取得します。

取得したライブラリは、「C:\iDKDoJa5.0」にインストールした場合、「C:\iDKDoJa5.0\bin」の下に置くだけで、i アプリオプションAPIを利用したアプリケーションを作成してエミュレーションできるようになります。

ライブラリに関する詳しい情報は、各メーカーのiアプリ関連サイトをご参照ください。各メーカーのiアプリ関連サイトへは、NTTドコモのページからたどることができます。

19. ヘルプからAPIリファレンスを参照したい場合は、別途、APIリファレンスを<http://www.nttdocomo.co.jp/service/imode/make/index.html> から取得し、インストールしたディレクトリのdocの下に置いてください。
（インストールしたディレクトリが「C:\iDKDoJa5.0」の場合、「C:\iDKDoJa5.0\doc」の下です。）

本書では、

「iα ppli Development Kit for DoJa-5.0(FOMA)」を「C:\iDKDoJa5.0」にインストールした場合として説明します。

ワンポイント

インストール後、「iα ppli Development Kit for DoJa-5.0(FOMA)」のヘルプメニューから操作ガイド（本書：ユーザーズガイド）とリリースノート参照できるようになります（「5.2 9. ヘルプの設定」参照）。
もし、インストール後、操作ガイドおよびリリースノートが参照できない場合は、インストールしたディレクトリの直下にこれらのファイル（操作ガイド、リリースノート）を置いてください。

2.3. アンインストール手順

「iαppli Development Kit for DoJa-5.0(FOMA)」をアンインストールするには：

1. Windowsの『スタート』→『プログラム』→『iαppli Development Kit for DoJa-5.0(FOMA)』→『Remove iαppli Development Kit for DoJa-5.0(FOMA)』を実行します（図 2-17）。
（「アプリケーションの追加と削除」から削除しても構いません。）

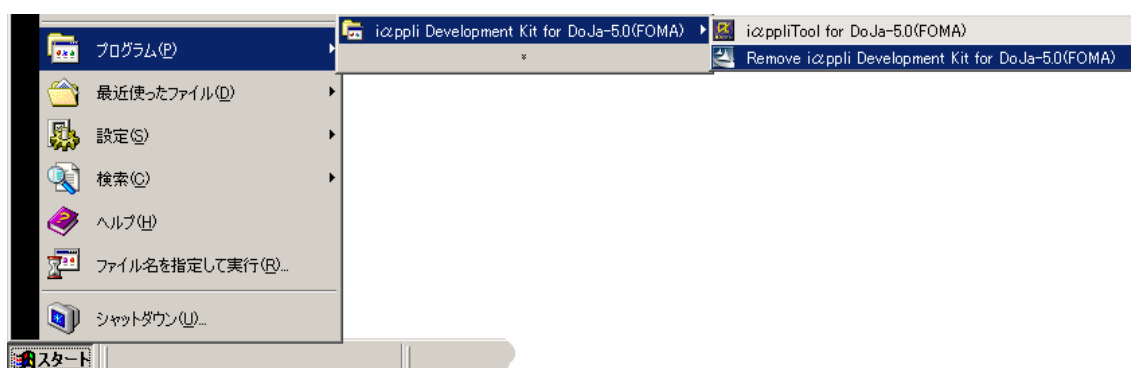


図 2-17 アンインストールの実行

2. 削除の確認画面が表示されるので、「OK」を押します（図 2-18）。

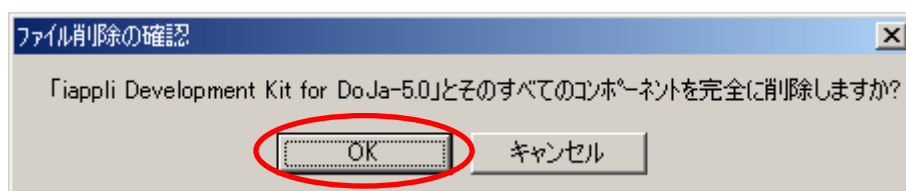


図 2-18 ファイル削除の確認画面

3. アンインストールを実行してもインストール先に作成された i アプリやデータは、そのまま残ります。
もし、それらも不要の場合は、インストール先のディレクトリ「C:\iDKDoJa5.0」を直接削除してください。

第3章. エミュレータの起動

エミュレータ (iαppliTool) を起動するには、Windowsの『スタート』→『プログラム』→『iαppli Development Kit for DoJa-5.0(FOMA)』→『iαppliTool for DoJa-5.0(FOMA)』を実行します。^[2]

エミュレータ (iαppliTool) が起動されると、図 3-2、図 3-1 に示す最初の画面が表示されます。

図 3-2 は、「エミュレータ (iαppliTool)」画面であり、i アプリのビルドや実行、携帯端末の設定などが行えます。

図 3-1 は、「エミュレーション端末」画面であり、ここで i アプリを実行します。この画面がエミュレータの待ち受け画面となります。

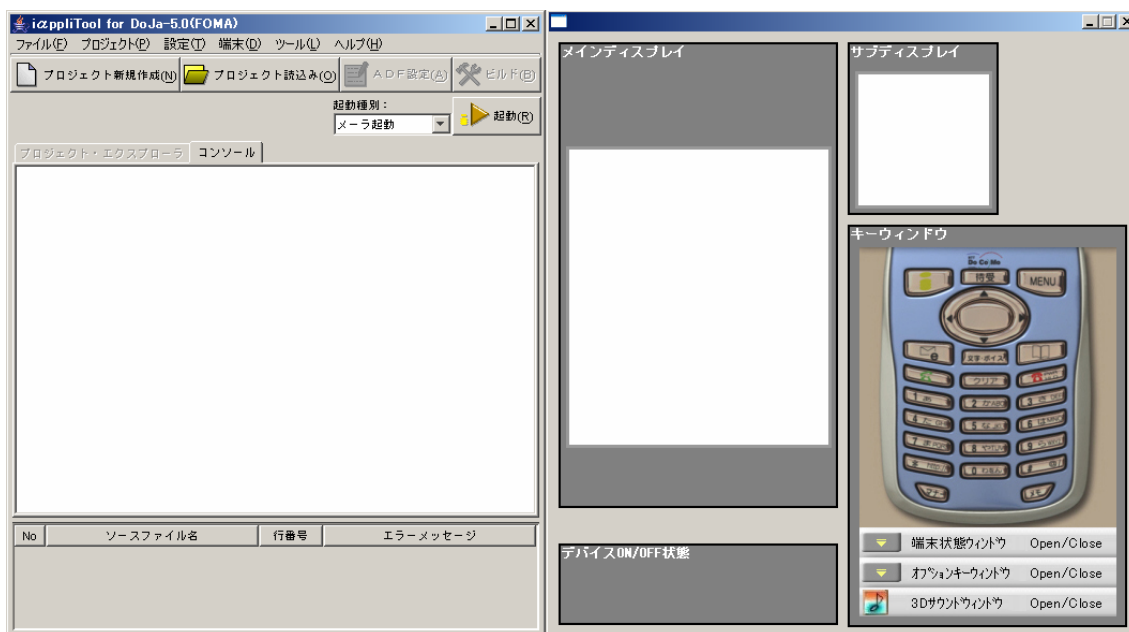


図 3-2 エミュレータ (iαppliTool) 画面



図 3-1 エミュレーション端末

^[2] エミュレータでプロジェクトのビルド（「4.3. プロジェクトのビルドと通常起動」参照）を実行したとき、エミュレータが J2SE SDK を見つけられない場合があります。この場合、「環境変数 JDK_HOME に J2SE SDK をインストールしたディレクトリを設定してください」というメッセージが表示されますので、まず環境変数 JDK_HOME に J2SE SDK をインストールしたディレクトリが正しく設定されているかを確認してください。

3.1. ビルドエラービュー

コンソールの下に表示されているビューで、アプリケーションのビルド時に発生したエラーを一覧表示します。表示されたエラーをダブルクリックすると、該当するプログラムをソースエディタで開くことができます（「4.2.3. ソースエディタの設定」参照）。

ソースエディタのパラメータに行番号を指定しておくと、エラー個所をすぐに表示できて、プログラムの修正が楽におこなえます。

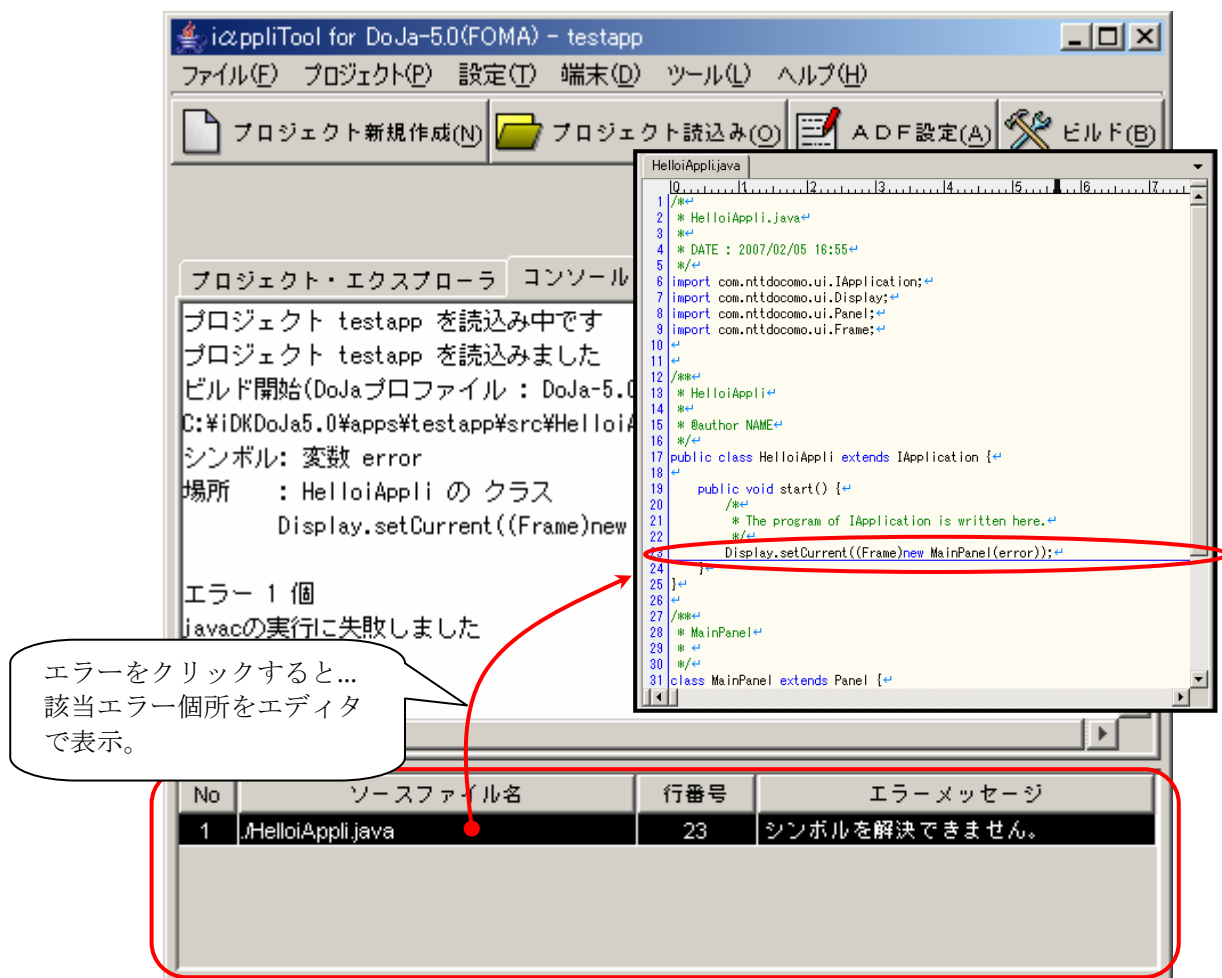


図 3-3 ビルドエラービュー

3.2. プロジェクト・エクスプローラ

プロジェクト・エクスプローラタブを選択すると、現在開いているプロジェクトのフォルダをツリー階層で表示します。

ここでは、以下の操作をおこなえます。

- ファイルをドラッグ&ドロップでコピーしたり、ファイルの名前の変更や削除をすることができます。
- 拡張子がtxtやjavaのファイルをダブルクリックすると、ソースエディタで開かれます（「4.2.3. ソースエディタの設定」参照）。
- Jamファイルをダブルクリックすると、ADF設定画面で内容を変更する事ができます（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。
- プロパティファイルをダブルクリックすると、プロパティ設定画面で内容を変更する事ができます（「4.5. プロジェクトの設定」参照）。
- プロジェクト直下にあるプロジェクト名のファイルをダブルクリックすると、トラステッド動作設定画面で内容を変更する事ができます（「5.2. トラステッド動作の設定」参照）。
- スクラッチパッドファイルをダブルクリックすると、スクラッチパッド管理ツールで内容を変更する事ができます（「6.1. スクラッチパッド管理ツール」参照）。
- 仮想FeliCaデータファイルをダブルクリックすると、仮想FeliCaデータファイルビューワで内容を変更する事ができます（「6.3. 仮想FeliCaデータファイルビューワ」参照）。
- component.list ファイルをダブルクリックすると、i アプリライブラリの選択画面で内容を変更する事ができます（「6.7. i アプリライブラリ」参照）。
- 右クリックの「開く」は、Windowsの関連付けに対応しています（図 3-5）。
- プロジェクトフォルダを右クリックすると「再読込」ができます。

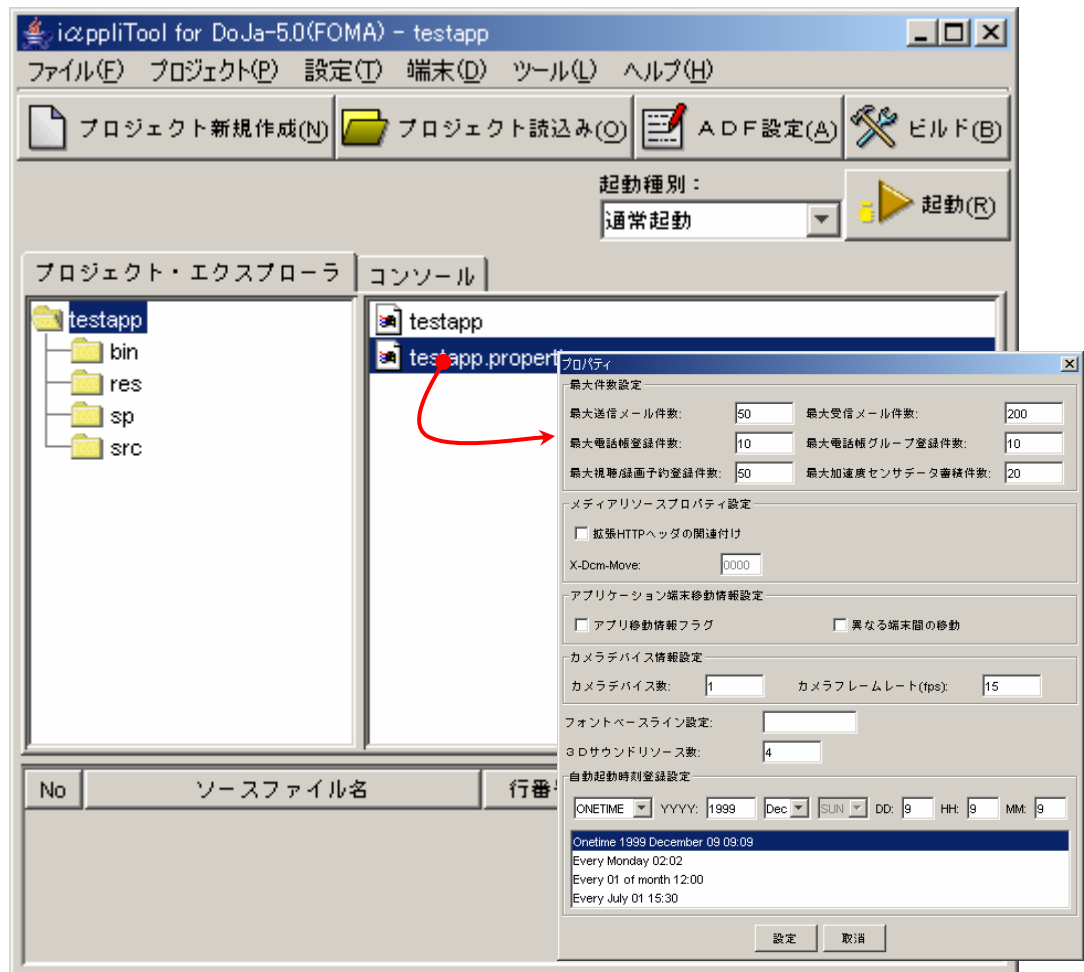


図 3-4 プロジェクト・エクスプローラ



図 3-5 ポップアップメニュー

第4章. iアプリ

この章では、iアプリの作成から実際に動かすところまでを説明します。

iアプリは、iモード携帯電話のJava環境で動作するアプリケーションです。iアプリには、通常に実行する**通常アプリ**、待ち受け画面で実行する**待ち受けアプリ**、実行が制限された**トラステッドiアプリ**があります（トラステッドiアプリについては「5.2.トラステッド動作の設定」を参照してください）。

iアプリをiモード携帯電話で動作させるには、最低でも次のものが必要になります：

- iアプリのJarファイル
- アプリケーションディスクリプタファイル（ADF）
- iアプリをダウンロードするためのHTMLファイル

これらは、すべてエミュレータで簡単に作成することができます。

4.1. iアプリの作成から起動まで

エミュレータでは、iアプリを「プロジェクト」という単位で作成します。「プロジェクト」とは、1つのiアプリに対するファイルの集合です。このプロジェクトには、ソースコードから最終的なJarファイルやアプリケーションディスクリプタファイル（ADF）、画像、音楽のリソースに至る全てのファイルが含まれています。

エミュレータでは、iアプリの作成から起動までを次の手順で行います：

- ① 新しいプロジェクトの作成（「4.2.プロジェクトの新規作成」を参照）
- ② iアプリの実行（「4.3.プロジェクトのビルドと通常起動」を参照）
- ③ 既存のプロジェクトの読み込み（「4.4.プロジェクトの読み込み」を参照）
- ④ プロジェクトの設定（「4.5.プロジェクトの設定」を参照）
- ⑤ iアプリの情報が記述されたアプリケーションディスクリプタファイル（ADF）の設定（「4.7.アプリケーションディスクリプタファイル（ADF）の編集」を参照）
- ⑥ プロジェクトのビルド

iアプリでは、スクラッチパッドという永続的な記憶領域を確保することができます。このスクラッチパッドにiアプリで動作させた情報を保存しておくことで、次回、起動した時にその情報を復元させることができます。スクラッチパッドは、各iアプリのアプリケーションディスクリプタファイル（ADF）のSPsizeを指定すると作成されます。

ワンポイント

iアプリのJarファイルは、ダウンロードが可能なサイズが限られています。そのため、あまり大きなJarファイルは、エミュレータで実行できていてもiモード携帯電話にダウンロードすることができません。

このような場合、iアプリで扱うリソースは、HTTP経由で取得してスクラッチパッドに保存することでJarファイルに含めないようにします。エミュレータに搭載されたサンプルでも、この手法を用いています。

4.2. プロジェクトの新規作成

新しいプロジェクトを作成するには：

1. メニューから「プロジェクト」→「新規作成...」を選択するか、「プロジェクト新規作成」ボタンを押します。

「プロジェクト新規作成画面」が表示されます（図 4-1）。



図 4-1 プロジェクト新規作成画面

2. プロジェクトの格納先（C:\iDKDoJa5.0\apps）を変更する場合は、「参照...」ボタンを押して格納先のディレクトリを選択します。
3. プロジェクト名を入力し、「作成」ボタンを押します。^[3]
「引き続きソースファイルを作成する」のチェックを外すとプロジェクトを作成するのみで閉じられます。

プロジェクトが作成されるとコンソール上にプロジェクトのソースコード（.javaファイル）とリソースファイル（アイコンなどのイメージファイルなど）、スクラッチパッドの格納場所を示すメッセージが表示されます（図 4-2）。

^[3] 新規プロジェクトを作成した際に、エミュレータは以下のディレクトリを作成します：

ソースコードの格納場所: C:\iDKDoJa5.0\apps\<プロジェクト名>\src
リソースファイルの格納場所: C:\iDKDoJa5.0\apps\<プロジェクト名>\res
スクラッチパッドの格納場所: C:\iDKDoJa5.0\apps\<プロジェクト名>\sp
Jar、ADF、HTML の格納場所: C:\iDKDoJa5.0\apps\<プロジェクト名>\bin

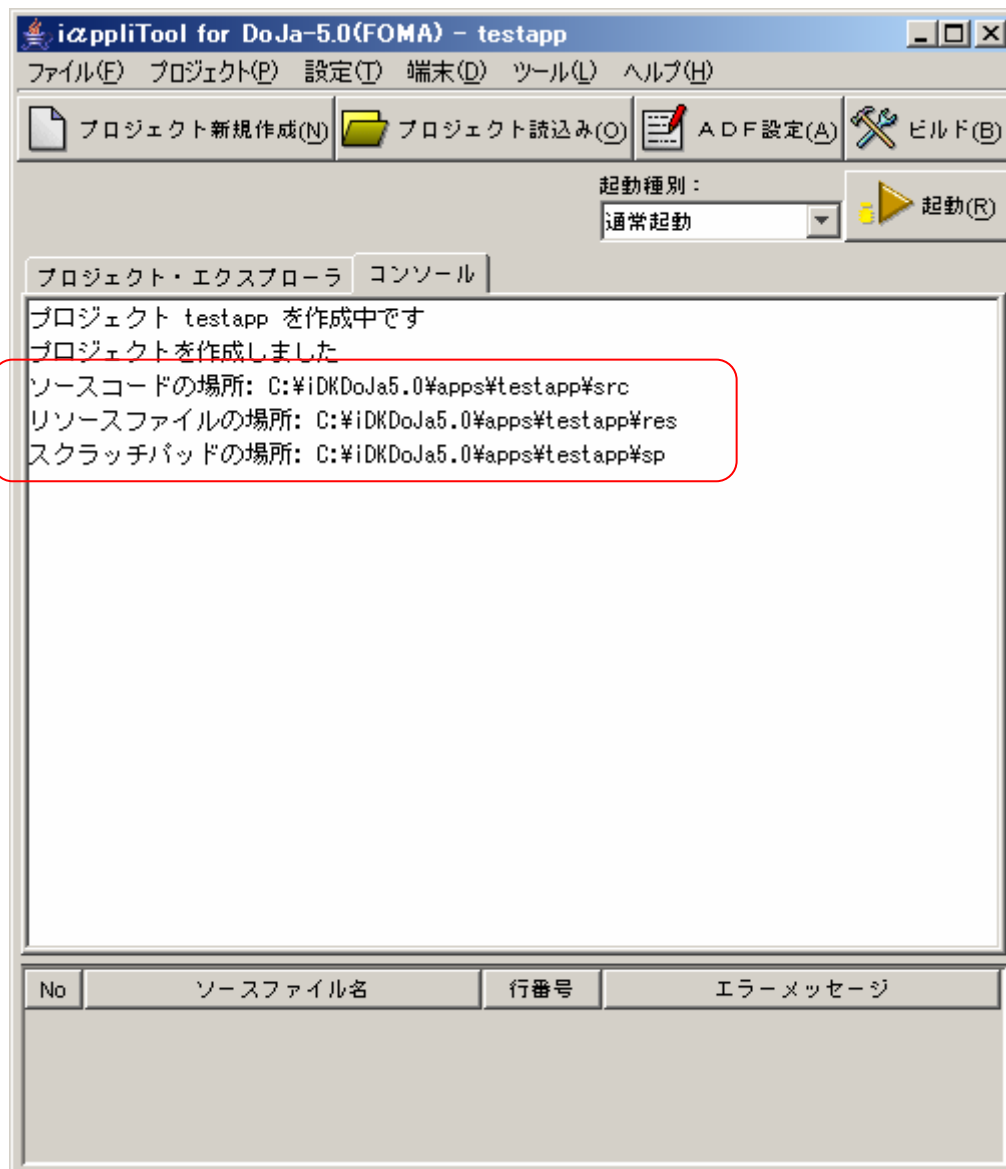


図 4-2 新規プロジェクト作成後のメイン画面

4. ソースファイルの新規作成画面が表示されます。ソースファイルの作成については、次の「4.2.1. ソースファイルの作成」を参照してください。

4.2.1. ソースファイルの作成

ソースファイル（ソースコードを記述したファイル）を作成するには：

1. メニューから「プロジェクト」→「ソースファイルの新規作成...」を選択します（図 4-3）。
前述のプロジェクトを作成する手順からでも同様です。

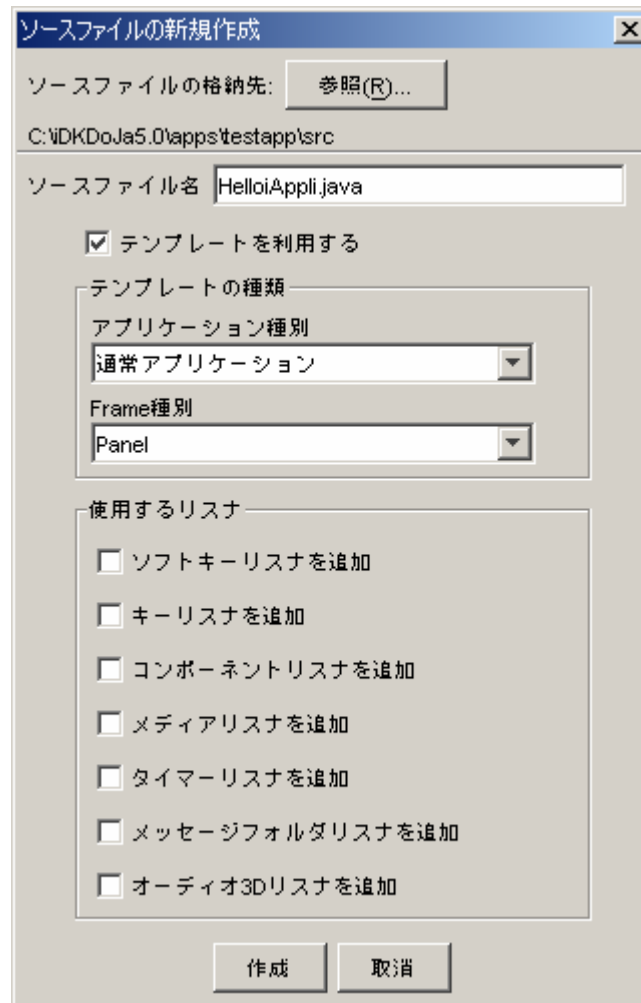


図 4-3 ソースファイルの新規作成画面

2. ソースファイルの格納先（C:\iDKDoJa5.0\apps\<プロジェクト名>\src）を変更する場合は、「参照...」ボタンを押して格納先のディレクトリを選択します。
通常、変更する必要はありません。
3. ソースファイル名を入力します。
4. 「テンプレートを利用する」にチェックを付けると、テンプレートのソースコードが書かれたソースファイルが作成されるようになります。

テンプレートを利用する場合：

- (1) テンプレートの種類「アプリケーション種別」で、
- 「通常アプリケーション (IApplication)」
 - 「待ち受けアプリケーション (MApplication)」
 - 「なし」
- のいずれかを選択します。^[4]

「なし」は、Canvas か Panel だけのクラスを作成する場合に選択します。
待ち受けアプリケーションに関しては、後ほど「4.8. 待ち受け起動」で説明します。

- (2) テンプレートの種類「Frame 種別」で、
- 「Panel」 (MainPanel というクラスで作成されます。)
 - 「Canvas」 (MainCanvas というクラスで作成されます。)
- のいずれかを選択します。



Panel は DoJa の高レベル A P I 用、Canvas は DoJa の低レベル A P I 用です。
ボタンやラベル、テキスト入力には、Panel。
画像や点/線の描画には、Canvas を用いるとよいでしょう。

^[4] テンプレートは、以下のファイルに格納されています。
これらは、自由に変更しても構いません。

- Panel 用通常アプリケーション：
C:\idKDoJa5.0\lib\template\IApplicationPanel.txt
- Panel 用待ち受けアプリケーション：
C:\idKDoJa5.0\lib\template\MApplicationPanel.txt
- Canvas 用通常アプリケーション：
C:\idKDoJa5.0\lib\template\IApplicationCanvas.txt
- Canvas 用待ち受けアプリケーション：
C:\idKDoJa5.0\lib\template\MApplicationCanvas.txt
- Panel のみ：
C:\idKDoJa5.0\lib\template\Panel.txt
- Canvas のみ：
C:\idKDoJa5.0\lib\template\Canvas.txt
- 各リスナの設定ファイル (埋め込みの定義)：
C:\idKDoJa5.0\lib\template\listener\xxxListener.conf

テンプレートファイル中で変換される変数は以下のとおりです。

変数名	変換内容
{CLASSNAME}	クラス名
{DATE}	作成日付
%IMPORT%	import 文の埋め込み
%IMPLEMENTS%	implements 文の埋め込み
%INIT_PROC%	コンストラクタの埋め込み
%ABSTRACT_METHOD%	リスナのメソッドの埋め込み
%FIELD%	フィールドの埋め込み

% で囲まれた埋め込み変数は変更しないでください。

(3) 使用するリスナで、使いたいリスナにチェックを付けます。

- ソフトキーリスナ . . . ソフトキーを扱う場合
- キーリスナ . . . キーを扱う場合
- コンポーネントリスナ . . . ボタンなどのコンポーネントを扱う場合
- メディアリスナ . . . 動画像や音のメディアデータを扱う場合
- タイマリスナ . . . タイマを扱う場合
- メッセージフォルダリスナ . . . メッセージフォルダを扱う場合
- オーディオ3Dリスナ . . . 3D音源を扱う場合

5. 「作成」ボタンを押します。

6. i アプリで参照するリソースファイル（画像や音楽など）があれば、それらを置いておきます。

i アプリのJarに含めてしまうもの：

- ・ リソースファイル：**C:\iDKDoJa5.0\apps\<プロジェクト名>\res** の下

あらかじめ i モード携帯電話上に保存されているもの：

- ・ マルチメディアファイル：**C:\iDKDoJa5.0\lib\multimedia** の下
（詳しくは、「10.9.マルチメディア領域」参照）

7. 任意のエディタでソースファイルが開かれるので、ソースコードを作成してください。
（任意のエディタについては、「4.2.3. ソースエディタの設定」参照）

（例）

図 4-3 でソースファイル名に "HelloiAppli" と入力し、テンプレートに「通常アプリケーション」の「Panel」を選択して作成すると、図 4-4 のようにファイルが開かれます。

このソースコードは、

C:\iDKDoJa5.0\apps\<プロジェクト名>\src\HelloiAppli.java
で作成されています。

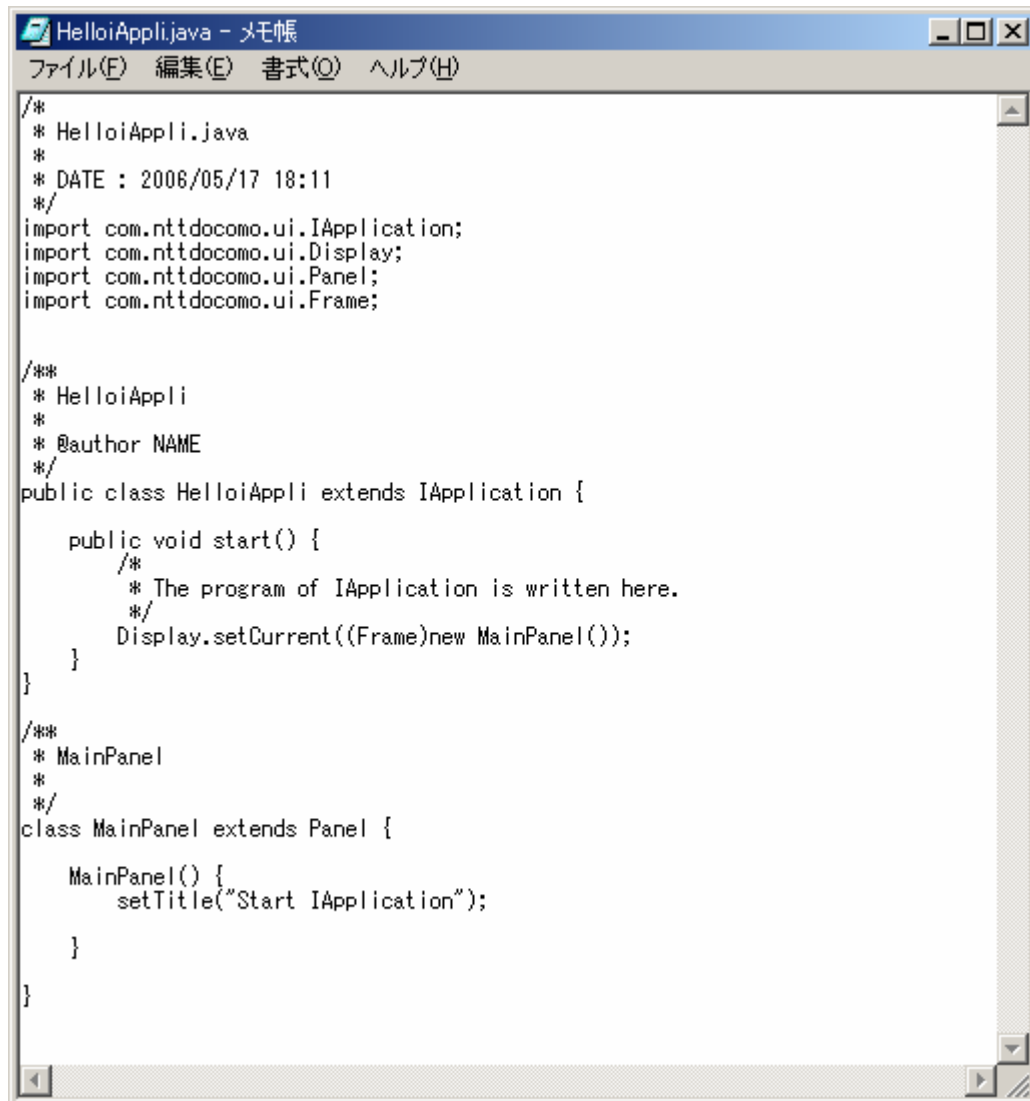


図 4-4 (例) HelloiAppli ソースコード

※1

ソースコードを記述するには、D o J a の A P I を理解する必要があります。
ソースコードを作成するために必要なD o J a の A P I リファレンスは、
NTTドコモの次のURLから入手することができます：
<http://www.nttdocomo.co.jp/service/imode/make/index.html>

4.2.2. ソースコードの修正

ソースコードを修正するには：

1. メニューから「プロジェクト」→「ソースファイルを開く...」を選択します。
ソースファイルは、直接、他のエディタで開いて変更しても構いません。
最終的に src の下に置かれたソースコードがコンパイルされます（「4.3. プロジェクトのビルドと通常起動」参照）。

ソースファイルの選択画面が表示されます（図 4-5）。

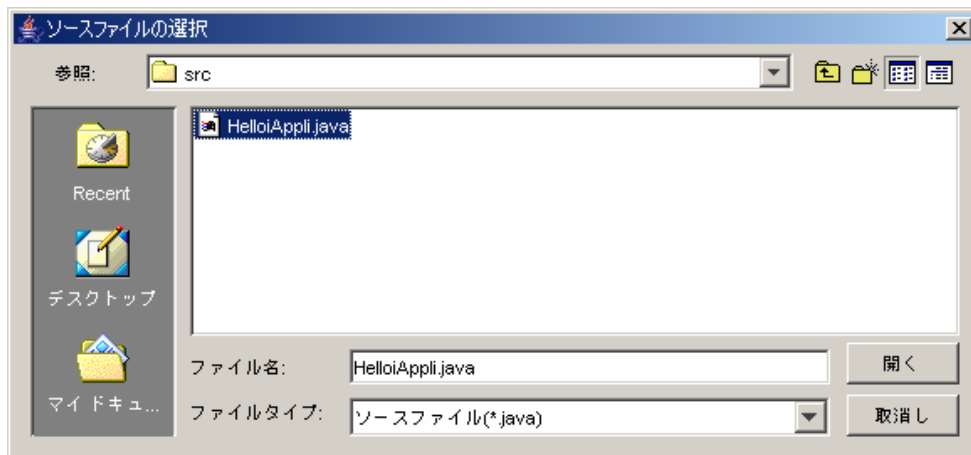


図 4-5 ソースファイルの選択画面

2. 修正したいソースファイルを選択して、「開く」を押します。
 3. 任意のエディタでソースファイルが開かれるので、ソースコードを作成してください（任意のエディタについては、「4.2.3. ソースエディタの設定」参照）。
- 4.2.1. ソースファイルの作成の 図 4-4 （例）HelloiAppli ソースコードで作成したソースコードの場合、図 4-6 のように開かれます。

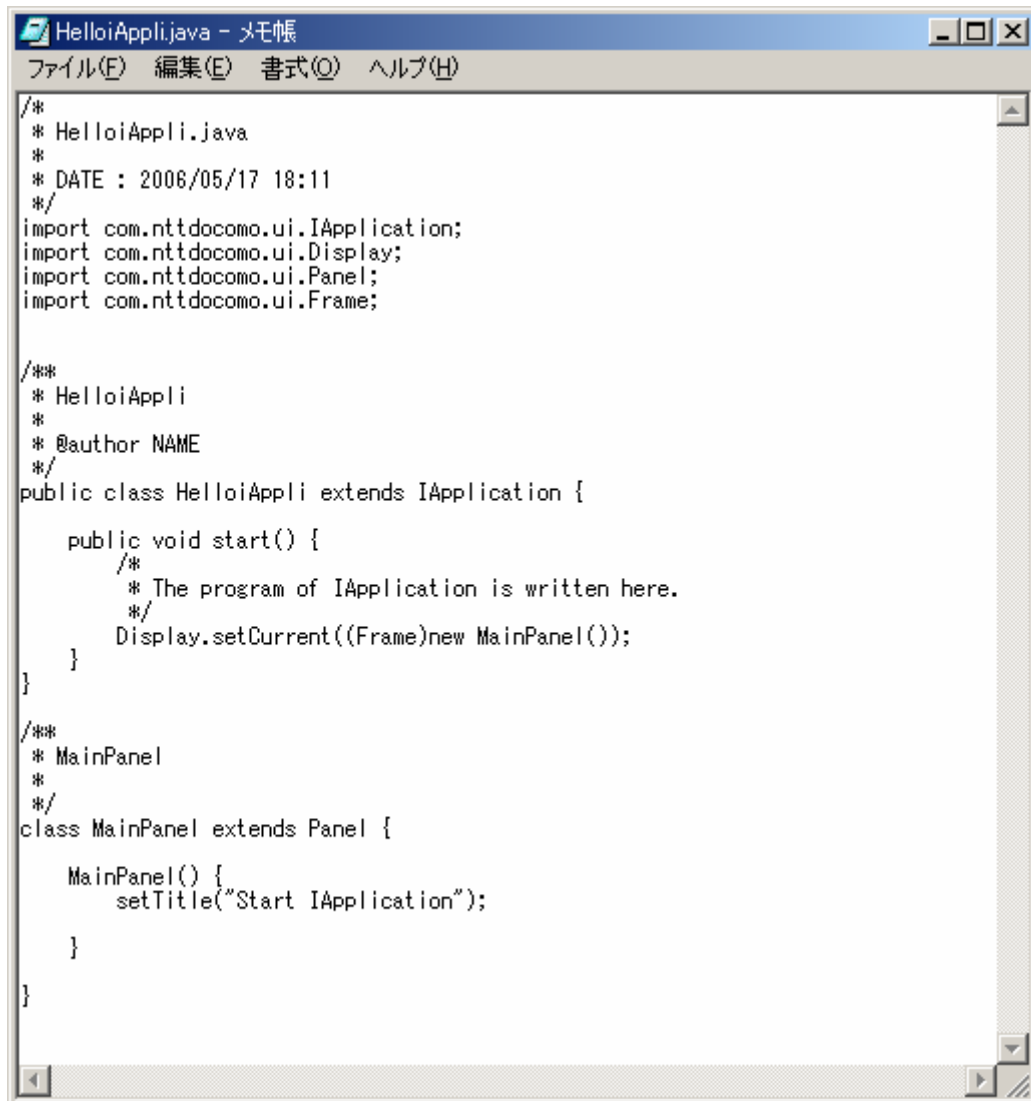


図 4-6 エディタで開かれたソースコード

4.2.3. ソースエディタの設定

ソースコードを編集するためのエディタは、初期値で「NOTEPAD.EXE」になっています。

ソースコードを編集するためのエディタを変更したい場合は：

1. メニューから「設定」→「エミュレータ環境設定...」→「ソースエディタの設定」タブを選択します。（図 4-7）

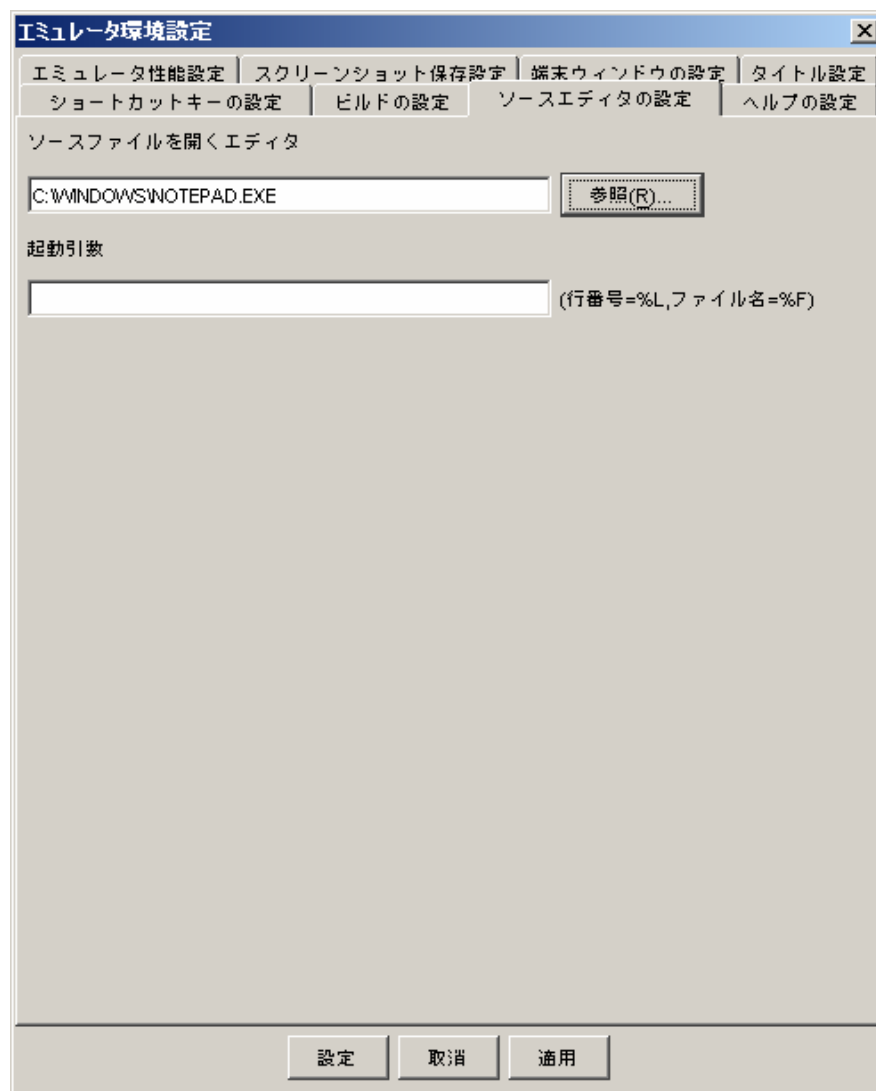


図 4-7 ソースエディタの設定画面

2. 「ソースファイルを開くエディタ」に任意のエディタの起動プログラムを指定するか、「参照...」を押してファイル選択ダイアログで選択します。
3. 必要に応じて起動引数を設定します。（※）
起動引数として起動先エディタの仕様に沿ったパラメータを指定します。
パラメータは、%L が行番号、 %F がファイル名に置き換わります。

なお、起動引数が指定された場合は設定内容をそのままエディタに渡すため、ソースファイル名（%F）の設定は必須となります。（ソースファイル名の自動付与は行いません。）

※本設定はビルドエラービューからのソースエディタ連携起動時のみ有効となります。その他の機能からの連携起動時は本項目の設定内容は無視され、自動的にソースファイル名を起動引数として付与し、エディタを起動します。
また、起動引数未設定時のビルドエラービューからの連携起動時につきましても、上記と同様に自動的にソースファイル名を起動引数として付与し、エディタを起動します。

4. 「設定」を押します。

ソースファイルがここで指定されたエディタで開かれるようになります。
ビルドエラービューからのソースエディタ連携時は、コンソールに表示されたビルドエラーのエラー行を選択すると、該当ソースファイルがここで設定したエディタで起動します。

この時、起動先エディタが行番号ジャンプに対応しており、かつ適切な起動引数の設定を行っている場合、ソースファイル中の該当エラー行が表示されます。

4.3. プロジェクトのビルドと通常起動

プロジェクトのビルドと通常起動を行うには：

1. メニューから「プロジェクト」→「ビルド」を選択するか、「ビルド」ボタンを押します（図 4-8）。

ソースコードのコンパイルと事前検証を行い、Jarファイル、アプリケーションディスクリプタファイルおよびダウンロード用のHTMLファイルを作成します。^[5]

このとき、アプリケーションの実行に必要なADFのAppNameとAppClassが未設定だった場合、ソースファイルから最初に見つかったIApplicationの実装クラス名が設定されます。アプリケーションディスクリプタファイル（ADF）については、「4.7. アプリケーションディスクリプタファイル（ADF）の編集」で説明します。

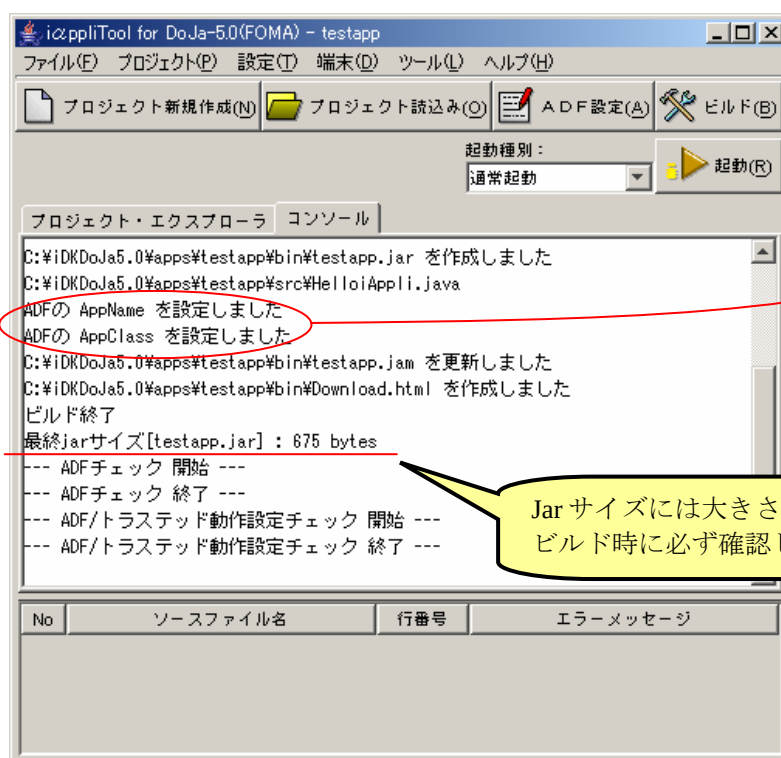


図 4-8 プロジェクトのビルド

2. メニューから「プロジェクト」→「ADF設定...」を選択するか、「ADF設定」ボタンを押します（図 4-9）。
アプリケーションに必要なADF項目を設定し、「設定」ボタンを押します（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

^[5] 正確には、既に存在する<プロジェクト名>.jam という ADF の内容を更新します（「AppSize」と「LastModified」属性が設定されます）。また、「ビルド」時に「環境変数 JDK_HOME に J2SE SDK をインストールしたディレクトリを設定してください」というメッセージが出力された場合、環境変数 JDK_HOME に J2SE SDK をインストールしたディレクトリを設定した後、改めて「iαppliTool」を起動してください。

ADF設定

AppName HelloiAppli

AppVer

PackageURL HelloiAppli.jar

AppSize 675

ConfigurationVer

ProfileVer

SPsize

AppClass HelloiAppli

AppParam

LastModified Thu, 01 Feb 2007 21:21:35

UseNetwork ☐ http

TargetDevice

LaunchAt

MyConcierge ☐ yes

UseTelephone ☐ call

UseBrowser ☐ launch

UseDTV ☐ launch

設定 取消

図 4-9 ADF の設定

(図 4-4 の例の場合は、AppName と AppClass が "HelloiAppli" になっています。
この例では特に変更する必要はありません。)

3. メニューから「プロジェクト」→「起動...」を選択するか、「起動」ボタンを押します。

ビルドした i アプリが起動されます (図 4-10)。

ただし、エミュレータが以下の状態の時には i アプリを起動できません：

- メーラ (「4.16. メーラ起動」を参照) が起動している
- ブラウザ (「4.17. ブラウザ起動」を参照) が起動している
- コードリーダー (「4.18. コードリーダー起動」を参照) が起動している

- トルカ（「4.2 3.トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

「通常起動」の起動タイプは、「LAUNCHED_FROM_MENU」です。
デフォルトの起動方法は「通常起動」ですが、その他にもいくつかの起動方法があります。



図 4-10 iアプリ 起動

4.3.1. ダウンロード用 HTML ファイル

i アプリをビルドすると、Jarファイル、アプリケーションディスクリプタファイルと同じ場所に、i アプリをダウンロードするためのHTMLファイル（Download.html）が作成されます（図 4-1 1）。

ただし、このファイルを直接書き換えた場合は、次回から再作成されなくなります。

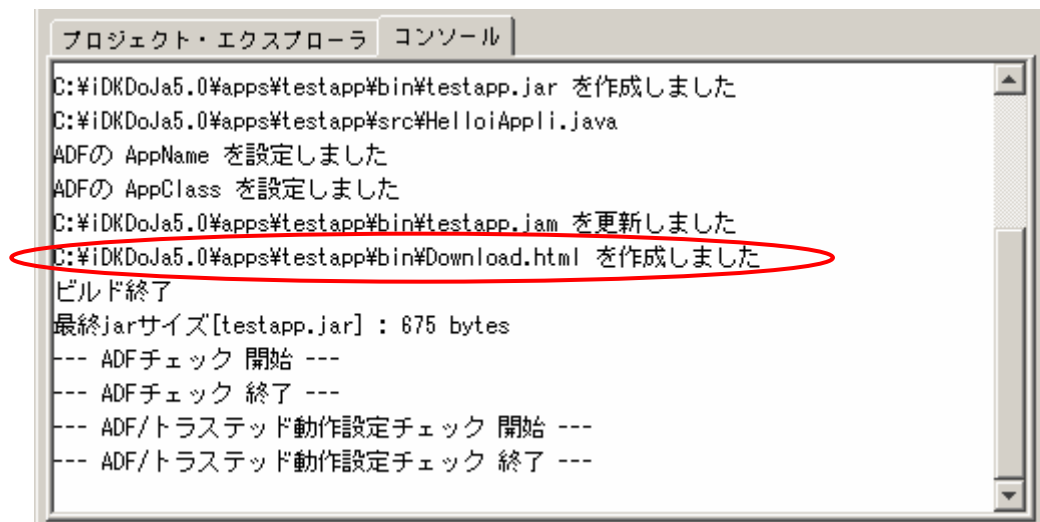


図 4-1 1 ダウンロード HTML ファイル作成

このHTMLファイルは、テンプレートから作成されています。
テンプレートとなるファイルは次の場所に置いてあります：

C:\iDKDoJa5.0\lib\template\Download.txt

また、テンプレートファイル中には、以下の定数を指定できます。

定数	内容
{PROJECTNAME}	プロジェクト名です。
{PACKAGEURL}	ADFのPackageURLの値と同じパスの jamファイルの位置です（4.7. アプリケーションディスクリプタファイル（ADF）の編集 参照）。PackageURLが設定されていない場合、プロジェクト名.jam になります。
{ATTRIBUTE}	起動種別により、ijam、ilet、ista のいずれかになります。 ijam … 即起動、ブラウザ起動以外の場合 ilet … 即起動の場合 ista … ブラウザ起動の場合

HTMLファイルのタグ記述については、「1 0.4. ブラウザ起動HTMLファイルの記述方法」を参照してください。

4.3.2. iモード携帯電話でiアプリ実行

もっとも簡単に作成した i アプリを i モード携帯電話上で実行する手順を説明します。

1. アプリケーションディスクリプタファイル (ADF) の「PackageURL」を、i アプリ (Jar) を配置する HTTP サーバの URL に変更します。(ADF については、「4.7. アプリケーションディスクリプタファイル (ADF) の編集」参照)
2. 「ビルド」を実行して、Jar、ADF、ダウンロード用 HTML を作り直します。
もし、ダウンロード用 HTML に手を加えている場合は再作成されませんので、直接、ダウンロード用 HTML 中の URL を修正してください。
3. 次のファイルを HTTP サーバ上に配置します。
 - C:\iDKDoJa5.0\apps\<プロジェクト名>\bin\<プロジェクト名>.jar
 - C:\iDKDoJa5.0\apps\<プロジェクト名>\bin\<プロジェクト名>.jam
 - C:\iDKDoJa5.0\apps\<プロジェクト名>\bin\Download.html
4. i モード携帯電話のブラウザから Download.html を開きます。
5. 「Download」リンクをクリックして、i アプリをダウンロードします。
6. i アプリを実行します。

ワンポイント

i アプリを i モード携帯電話で動作させるとき、ダウンロードした機種を特定させたい場合があります。

i モード携帯電話の機種を特定させるには、次の2つの方法があります。

1 つは、`java.lang.System.getProperty("microedition.platform")` を使って i アプリが取得する方法 (エミュレータでは、ネットワーク設定の **UserAgent に設定された "DOJA_INET_CLIENT"** の部分を返します)、もう 1 つは、HTTP 通信時に i モードサーバから送出される User-Agent ヘッダを見る方法です。

(詳しくは、「i アプリコンテンツ開発ガイド」を参考にしてください。)

4.4. プロジェクトの読み込み

既存プロジェクトを読み込むには：

1. メニューから「プロジェクト」→「読み込み…」を選択するか、「プロジェクト読み込み」ボタンを押します。

「プロジェクト読み込み」画面が表示されます（図 4-1 2）。^[6]



図 4-1 2 プロジェクト読み込み画面

2. プロジェクトの格納先（C:\iDKDoJa5.0\apps）を変更する場合は、「参照…」ボタンを押して格納先のディレクトリを選択します。
一覧のタイトル「名前」、「更新日時」をクリックすると並び替えて表示されます。

^[6] プロジェクト一覧の「情報」欄は、デフォルトでADFの AppName の値が表示されます。（ADFについては、「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）

この情報を任意の説明に変更したい場合は、規定クラスのドキュメンテーションコメントに「@info」で付与します。（ただし、画面には最大 128 バイトまでしか表示されませんので、i アプリの概要が分かる程度の短い文章で記述しましょう。）

<例>

```
/**
 * @info HelloiAppli のメインクラスです。
 *
 * @author NAME
 */
public class HelloiAppli extends IApplication {
    :
```


3. プロジェクト名を選択し「読み込み」ボタンを押します。

メイン画面のタイトルバーにプロジェクト名が表示されます（図 4-1 3）。

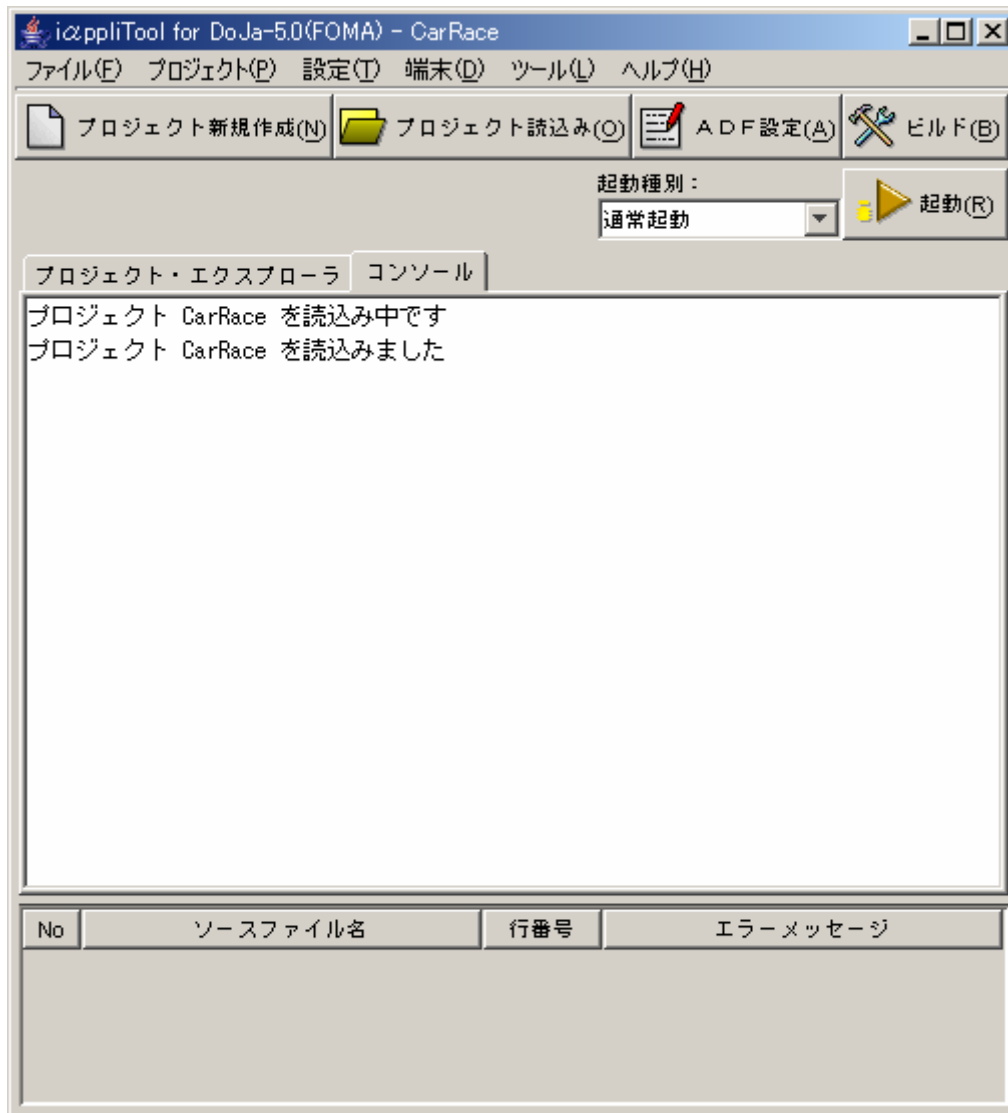


図 4-1 3 既存プロジェクト読み込み後のメイン画面

4. 「起動」ボタンを押すと読み込んだプロジェクトの i アプリが実行されます。

・プロジェクト読み込み画面に表示されるアイコンについて

プロジェクト読み込み画面に表示されるアイコンは、ADFのAppIconに指定したアイコンが表示されます。（ADFについては、「4.7.アプリケーションディスクリプタファイル（ADF）の編集」参照）

AppIconのアイコンは、48×48 または、96×96 のサイズで作成された GIF 形式のファイルです。このアイコンは、res 配下に置きます。

もしアイコンが正しくない場合は、× 表示されます（図 4-1 4）。



図 4-1 4 アイコン表示

4.5. プロジェクトの設定

i モード携帯電話に合わせたプロジェクトの設定を行います。

プロジェクトの設定をするには：

1. メニューから「プロジェクト」→「プロジェクト設定...」を選択します（図 4-15）。

プロジェクト設定

最大件数設定

最大送信メール件数: 50 最大受信メール件数: 200

最大電話帳登録件数: 10 最大電話帳グループ登録件数: 10

最大視聴/録画予約登録件数: 50 最大加速度センサデータ蓄積件数: 20

メディアリソースプロパティ設定

☐ 拡張HTTPヘッダの関連付け

X-Dcm-Move: 0000

アプリケーション端末移動情報設定

☐ アプリ移動情報フラグ ☐ 異なる端末間の移動

カメラデバイス情報設定

カメラデバイス数: 1 カメラフレームレート(fps): 15

フォントベースライン設定:

3Dサウンドリソース数: 4

自動起動時刻登録設定

ONETIME YYYY: 1999 Dec SUN DD: 9 HH: 9 MM: 9

Onetime 1999 December 09 09:09

Every Monday 02:02

Every 01 of month 12:00

Every July 01 15:30

設定 取消

図 4-15 プロジェクトの設定

2. 必要な設定項目を設定します。
3. 「設定」ボタンを押すと、変更が有効になります。
各設定項目は、表 4-1 に示すとおりです。

表 4-1 プロジェクトの設定項目

設定項目	説明	デフォルト
最大送信メール件数	送信/未送信メールの最大保存件数を設定します。	50
最大受信メール件数	受信メールの最大保存件数を設定します。	200
最大電話帳登録件数	電話帳エントリの最大保存件数を設定します。	10
最大電話帳グループ登録件数	電話帳グループの最大保存件数を設定します。	10
最大視聴/録画予約登録件数	視聴/録画予約の最大保存件数を設定します。	50
最大加速度センサデータ蓄積件数	加速度センサデータの最大保存件数を設定します。	20
拡張 HTTP ヘッダの関連付け	メディアに設定する HTTP のヘッダ値 "X-Dcm-Move" で、「0000」か「0001」を設定します。	設定なし
アプリ移動情報フラグ	IApplication#isMoved() で取得できる状態を設定します。	チェックなし (false)
異なる端末間の移動	IApplication#isMovedFromOtherTerminal() で取得できる状態を設定します。	チェックなし (false)
フォントベースライン設定	文字のベースラインを変更する場合に設定します。	なし
カメラデバイス数	カメラの数を設定します。 Camera#getCamera() のカメラ ID 0, 1, 2... の順で対応します。	1
カメラフレームレート	カメラのフレームレートを設定します。(fps)	15
3Dサウンドリソース数	Audio3D#getResources() で取得できる 3D サウンドのリソース数を設定します。	4
自動起動時刻設定 自動起動時刻一覧	自動起動時刻を設定します。 ※ 値を変更して別の項目に移動すると一覧に反映されます。	Onetime 1999 January 09 09:09 Every Monday 02:02 Every 01 of month 12:00 Every July 01 15:30

詳細については、「10.3.1. 「アプリケーション名.properties」ファイル」を参照してください。

4.6. ファイル配置カスタマイズ

ビルドされた i アプリを自動的に別の場所に配置したい場合、配置先を変更する事ができます。

i アプリの配置を設定するには：

1. メニューから「プロジェクト」→「ファイル配置カスタマイズ(D)...」を選択します（図 4-16）。

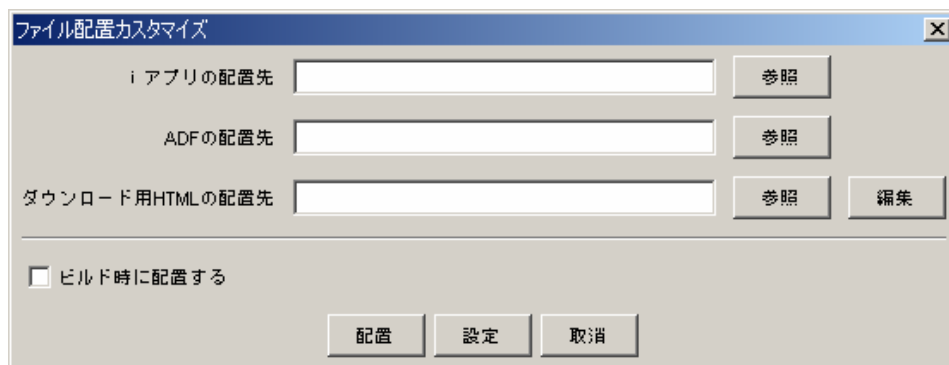


図 4-16 ファイル配置カスタマイズ画面

2. 必要な設定項目を設定します。
3. 「設定」ボタンを押すと、変更が有効になります。
各設定項目は、表 4-2 に示すとおりです。
4. 「配置」ボタンを押すと、それぞれのファイルを配置します。
5. 「ビルド時に配置する」にチェックをつけると、ビルドしたときに毎回配置されます。
これにより、「iα ppli Development Kit for DoJa-5.0(FOMA)」で作成された i アプリが、そのまま本番環境にも反映されている状態となります。

表 4-2 ファイル配置カスタマイズの設定項目

設定項目	説明	デフォルト
i アプリの配置先	i アプリ (xxx.jar) の配置先を設定します。 ファイル名まで指定するとそのファイル名に変更されます。	なし
ADF の配置先	ADF (xxx.jam) の配置先を設定します。 ファイル名まで指定するとそのファイル名に変更されます。	なし
ダウンロード用HTMLの配置先	ダウンロード用HTMLの配置先を設定します。 ファイル名まで指定するとそのファイル名に変更されます。	なし

4.7. アプリケーションディスクリプタファイル(ADF)の編集

i モード携帯電話では、Java Application Manager (JAM) と呼ばれるコンポーネントで i アプリの実行を管理しています。そのため、i アプリの Jar ファイルとは別に i アプリの情報を JAM に渡すためのファイル (アプリケーションディスクリプタファイル: ADF) が必要になります。

アプリケーションディスクリプタファイル(ADF)を編集するには:

1. メニューから「プロジェクト」→「ADF 設定...」を選択するか、「ADF 設定」ボタンを押します。
「ADF設定」画面が表示されます (図 4-17)。

ADF設定

AppName: CARレース

AppVer:

PackageURL: http://localhost/CarRace/bin/CarRace.jar

AppURL: [必須入力項目]
i アプリの格納場所を示すURL
プロジェクト作成時に自動設定

Configuration:

ProfileVer:

SPsize: 20,30700

AppClass: CarRace

AppParam:

LastModified: Thu, 29 Jun 2006 12:02:13

UseNetwork: ☒ http

TargetDevice:

LaunchAt:

MyConcierge: ☐ yes

UseTelephone: ☐ call

UseBrowser: ☐ launch

UseDTV: ☐ launch

設定 取消

各項目にマウスポインタを合わせるとツールチップで設定する内容が表示されます。

図 4-17 ADF 設定画面

2. 必要な設定項目を入力します。
3. 「設定」ボタンを押すと、変更が有効となります。
各設定項目は、表 4-3 に示すとおりです。

表 4-3 アプリケーションディスクリプタファイル(ADF)の設定項目

設定項目	説明	種別
AppName	i アプリの名称（最大 16 バイト）。設定する文字列は、SJIS を使用します。 プロジェクトのビルド時に未設定であれば、IApplication が実装されたクラス名が設定されます。 （ただし、IApplication がソースコード中に複数ある場合、最初に見つかったクラスになります。）	必須
AppVer	i アプリのバージョン（最大 10 バイト）。 この項目は、エミュレータでは使用しないので設定しなくても問題ありません。	オプション
PackageURL	i アプリの格納場所を表す URL（最大 255 バイト）。 設定する文字列は、ASCII 形式とします。ただし、サーバの IP アドレスを直接指定された場合は、アプリケーションの起動を禁止します。 この項目に相対URLを設定した場合、「5.1. ネットワークの設定」で設定した「ADFのURL」フィールドがベースのURLとなります。 i アプリはこの PackageURL のホストとポートのみ通信可能です。 新規プロジェクトを作成したときには、i アプリの Jar ファイル名になっているので、適切な URL（i アプリのダウンロード先）に変更してください。	必須
AppSize	アプリケーションサイズ（バイト単位）。 実際の i アプリのサイズ（Jar）と違っていたり、0 または負数が指定されたりした場合は、アプリケーションの起動を禁止します。 プロジェクトのビルド時に生成された Jar ファイルのサイズが設定されます。	必須
ConfigurationVer (KvmVer)	コンフィグレーション(KVM)のバージョン。 KvmVer と ConfigurationVer はどちらも同じ意味とします。 ただし、両方指定された場合は、アプリケーションの起動を禁止します。 設定する文字列は、“CLDC-1.0” (DoJa-1.0, 2.0, 2.1, 2.2, 3.0, 3.5) または、“CLDC-1.1” (DoJa-4.0, 4.1, 5.0) とし、それ以外を設定した場合は、i アプリの起動を禁止します。 但し、大文字・小文字の区別はしません。 この項目は、エミュレータでは使用しないので設定しなくても問題ありません。	オプション

ProfileVer	プロファイルのバージョン。 設定する文字列は、以下のとおりで、それ以外を設定した場合は、アプリケーションの起動を禁止します。但し、大文字・小文字の区別はしません。 この項目は、エミュレータでは使用しないので設定しなくても問題ありません。 ・"DoJa-1.0" (503i シリーズ対応) ・"DoJa-2.0" (504i シリーズ対応) ・"DoJa-2.1" (FOMA2051 シリーズ対応) ・"DoJa-2.2" (FOMA2102 シリーズ対応) ・"DoJa-3.0" (505i、506i シリーズ対応) ・"DoJa-3.5" (900i シリーズ対応) ・"DoJa-4.0" (901i シリーズ対応) ・"DoJa-4.1" (902i シリーズ対応) ・"DoJa-5.0" (903i/904i シリーズ対応)	オプション
SPsize	スクラッチパッドのサイズ (バイト単位)。 複数のスクラッチパッドは、カンマ(,)で区切って指定します。(最大16個) DoJa-5.0 プロファイル以降の仕様に従い、エミュレータ上でも Jar ファイルとスクラッチパッドの合計サイズに制限があります。この項目を設定しない場合のデフォルト値は0バイトとなります。 また、負数、整数以外が設定された場合は、アプリケーションの起動を禁止します。 (例: 16, 1024) スクラッチパッドは、アプリケーションディスクリプタファイルの編集画面 (図 4-17) で「設定」を押すと再作成(自動拡張)されます。このとき既に記憶されているファイル中のデータはサイズの範囲内でそのまま残ります。	オプション
AppClass	i アプリのメインクラス名 (最大 255 バイト)。 このクラス名は、com.nttdocomo.ui.IApplication のサブクラスの名前でなければなりません。 設定する文字列は、ASCII 形式とし、255 バイトを超えて設定した場合は、アプリケーションの起動を禁止します。 ----- プロジェクトのビルド時に未設定であれば、IApplication が実装されたクラス名が設定されます。 (ただし、IApplication がソースコード中に複数ある場合、最初に見つかったクラスになります。)	必須
AppParam	メインクラスに渡すパラメータ (最大 255 バイト)。 パラメータは、スペースで区切って設定します。アプリケーションは、com.nttdocomo.ui.IApplication#getArgs() メソッドを使用してこのパラメータを取得します。 このメソッドは、この項目が設定されていないと null を返します。設定できる文字列は ASCII 形式とし、255 バイト以内ならパラメータの個数に制限はありません。 しかし、255 バイトを超えて設定した場合は、アプリケーションの起動を禁止します。	オプション

LastModified	i アプリの最終更新時刻。 設定する文字列の形式については「4.7.3. LastModified 項目の表記方法」を参照。この項目は、ビルドのJarファイル生成時に自動的に設定されます。 プロジェクトのビルド時にビルドした時間が設定されます。	必須
UseNetwork	i アプリがネットワーク通信を使用することを表すフラグ。 有効な値は "http" (ASCII 形式とし、大文字・小文字の区別はしません) のみとし、"http" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しなかった場合は、ネットワーク接続できない i アプリとみなされますので、ネットワークを接続しようとする例外が発生します。	オプション
TargetDevice	i アプリが動作する機種名 (user agent に記載されているものと同等で、最大 128 バイト)。 この項目は、特定の機種に対してのみ i アプリのダウンロードを許可したいときに設定します。 設定する文字列は、ASCII 形式とし、大文字・小文字の区別はしません。カンマ(,)で区切られた文字列を 1 機種とします。 複数の機種をターゲットとしている場合、"X2005i, Y2005i, Z2005i" (PDC(MOVA)の場合、"X506i, Y506i, Z506i") のように設定します。 また、この項目に何も設定しなかった場合は、全ての機種をターゲットとしているとみなされます。 シリーズの後に "i", "is" 等が付与されている場合も同一シリーズとみなされてダウンロードできます。 同一シリーズ全機種をターゲットとする場合は、"all2005" のようにシリーズ番号の前に "all" を付加します。 この項目は、エミュレータでは使用しないので設定しなくても問題ありません。	オプション
LaunchAt	起動時間間隔指定。 i アプリを一定時間ごとに自動起動させるときに設定します。 形式: I h ("h"は起動間隔 (時間 単位)。I と h の間には 1 つのスペースを入れる。例: "I 6" ... 6 時間毎に起動) "h"に 0 が設定された場合は、無効なものとみなします。 "h"の有効桁数は 2 桁とします。 また、設定する文字列は、ASCII 形式とし、それ以外が設定された場合は、アプリケーションの起動を禁止します。但し、大文字・小文字の区別はしません。	オプション

MyConcierge	待ち受けアプリケーションかどうかを表すフラグ。 この項目は、エミュレータ上で待ち受け起動可能であるかどうかの判定に使用します（「4.8.待ち受け起動」参照）。 有効な値は "yes"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"yes" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、待ち受け起動しない i アプリであるとみなし、エミュレータ上で待ち受け起動を禁止します。	オプション
UseTelephone	電話の音声通話機能を利用するかどうかを表すフラグ。 有効な値は "call"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"call" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、電話の音声通話機能を利用しないアプリケーションであるとみなします。	オプション
UseBrowser	i アプリからブラウザ機能を利用するかどうかを表すフラグ。 有効な値は "launch"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"launch" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、ブラウザ機能を利用しないとみなしますので、アプリケーションからブラウザ機能を使用すると例外が発生します。	オプション
UseDTV	i アプリから DTV アプリを利用するかどうかを表すフラグ。 有効な値は "launch"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"launch" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、DTV 機能を利用しないとみなしますので、アプリケーションから DTV を起動すると例外が発生します。	オプション
LaunchByMail	メールから i アプリの起動を許可するメールアドレス（最大 50 バイト）。 この項目に "any" を設定した場合は、任意のメールアドレスからのアプリケーションの起動を許可します。 "any" 以外のメールアドレスを設定した場合は、アプリケーションを起動しようとしているメールアドレスの From: アドレスと、この項目で設定した値が完全に後方一致するときのみ起動を許可します。 それ以外はアプリケーションの起動を禁止します。 また、この項目に複数のメールアドレスを設定した場合は、先頭の 1 つのメールアドレスのみ有効とします。 設定する文字列は、ASCII 形式とします。	オプション

LaunchByBrowser	ブラウザから i アプリの起動を許可する URL（最大 255 バイト）。 この項目に"any"を設定した場合は、任意の URL からのアプリケーションの起動を許可します。 "any"以外の URL を設定した場合は、アプリケーションを起動しようとしている URL と、この項目で設定した値が完全に前方一致するときのみ起動を許可します。 また、この項目に複数の URL を設定した場合は、先頭の 1 つの URL のみ有効とします。 なお、先頭から空白、タブまでを 1 つの URL とみなします。 設定する文字列は、ASCII 形式とします。	オプション
LaunchByToruCa	トルカスニップから i アプリの起動を許可する URL（最大 255 バイト）。 この項目に"any"を設定した場合は、任意の URL からのアプリケーションの起動を許可します。 "any"以外の URL を設定した場合は、アプリケーションを起動しようとしている URL と、この項目で設定した値が完全に前方一致するときのみ起動を許可します。 また、この項目に複数の URL を設定した場合は、先頭の 1 つの URL のみ有効とします。 なお、先頭から空白、タブまでを 1 つの URL とみなします。 設定する文字列は、ASCII 形式とします。	オプション
LaunchByBML	BML ブラウザから i アプリの起動を許可する系列局 ID（10 進数表記で 0 から 11 までの値）。 i アプリケーションを起動しようとしている BML コンテンツの配信元となる系列局 ID と、この項目で指定された系列局 ID が一致するときのみ起動を許可します。 指定できる ID はひとつとし、先頭から数字以外の文字までを 1 つの ID と見なし、それ以降の文字は無視します。 設定する文字列は、ASCII 形式とします。	オプション

AllowPushBy	i アプリのプッシュ起動を許可する外部インタフェース（最大 255 バイト）。 この項目に "Irda:<起動コマンド>" を設定した場合は赤外線、 "Code:<起動コマンド>" を設定した場合はコードリーダー、 "icc:<起動コマンド>" を設定した場合は FeliCa の R/W から i アプリの起動を許可します（<起動コマンド>は、i アプリの起動チェックの際に使用し、1 文字以上の文字列とします）。 "Irda:"、"Code:"、"icc:" 以外の文字を指定した場合は、アプリケーションの起動を禁止します。 また、複数の外部インタフェースを設定する場合は、空白、タブで機器を区切ります。 この項目に複数の外部インタフェースを設定した場合は、先頭の 1 つの外部インタフェースのみ有効とします。	オプション
AppTrace	i アプリのトレース機能を利用するかどうかを表すフラグ。 有効な値は "on"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"on" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しなかった場合は、i アプリのトレース機能を使用しないものとみなします。 この項目は、エミュレータでは使用しないので設定しなくても問題ありません。	オプション
DrawArea	i アプリが使用する描画領域。 この項目に以下の形式に従った値を設定した場合: 形式: (横のピクセル数) x (縦のピクセル数) 96 <= (横のピクセル数) <= 端末の最大横ピクセル数 72 <= (縦のピクセル数) <= 端末の最大縦ピクセル数 i アプリ起動時の画面を、端末の描画領域の中央に設定します。（x は小文字の x (ASCII 0x78) です） その際、描画領域周囲の表示色は実装依存とし、i アプリから描画を行うことはできません。 上記の形式（設定可能なピクセル数は正の整数のみ）以外の値を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しなかった場合は、通常通りの描画とします。	オプション

GetSysInfo	メール、メッセージアイコン、圏外／圏内情報、電池残量、マナーモード設定情報取得可否を表すフラグ。 有効な値は "yes" (ASCII 形式とし、大文字・小文字の区別はしません) のみとし、"yes" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、メール、メッセージアイコン、圏外／圏内情報、電池残量、マナーモード設定情報を取得しないものとみなしますので、i アプリからこれらを取得しようとする場合例外が発生します。	オプション
TrustedAPID	NTTドコモで付与するトラステッドの管理ID。 この項目に11桁の数字以外が設定された場合は、アプリケーションの起動を禁止します。 この項目が設定された場合は、i アプリはトラステッド (i アプリ d x) であるとみなします。 <u>トラステッド動作設定のいずれかを "Yes" にした場合は、この項目に "100000000000" などの 11 桁の数字文字列を設定してください。</u>	オプション
TrustedLmd	トラステッドの最終更新日時	オプション
PackageHashCode	Jar ファイルの Hash 値。 PackageURL で指定された Jar の Hash 値 (SHA-1) を Base64 エンコードした値を設定します。 SD-Binding 機能を利用する際に CPBindingAuth キーを設定する必要がある場合は、本キーも共に設定する必要があります。 PackageHashCodeは、CPBindingAuthを設定するツールで追加する事ができます。CPBindingAuthの設定については、「6.8. CPBindingAuth署名付与ツール」を参照してください。	オプション
LaunchByApp	ランチャモードによる他の i アプリからの起動禁止を表すフラグ。 有効な値は "deny" (ASCII 形式とし、大文字・小文字の区別はしません) のみとし、"deny" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 この項目が設定されている場合は、ランチャモードにより他の i アプリから連携起動をしようとする場合例外が発生します。	オプション

MessageCode	NTTドコモで付与するメッセージiアプリの管理ID。 この項目に10桁のASCII形式の文字列以外が設定された場合は、アプリケーションの起動を禁止します。 また、この項目が設定されている場合は、メッセージiアプリであるとみなします。この項目は、メールフォルダや送受信メールの管理IDになります。 メッセージiアプリである場合は、この項目を設定し、トラステッド動作設定→メッセージiアプリ機能の許可をYes、TrustedAPIDに"0000000000"などの11桁の数字文字列を設定してください。	オプション
AccessUserInfo	電話帳登録・参照、スケジュール登録、スケジュール起動、画像登録・参照、動画登録・参照、ブックマーク登録、アラーム登録、デコメールテンプレート登録・参照、トルカデータ登録・参照、マイプロフィール参照の可否を表すフラグ。 有効な値は"yes"（ASCII形式とし、大文字・小文字の区別はしません）のみとし、"yes"以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、電話帳登録、スケジュール登録、マルチメディアへの画像保存／参照、ブックマーク登録／参照、デコメールテンプレート登録はしないものとみなしますので、アプリケーションからこれらの機能を実行しようとする例外が発生します。	オプション
GetUtn	端末及びUIMの固体識別番号の利用を表す文字列。この項目に設定された固体識別番号をアプリケーションで取得することができます。 UIMの固体識別番号を取得したい場合は、"userid" 端末の固体識別番号を取得したい場合は、"terminalid" 両方を取得したい場合は、"userid,terminalid" もしくは、"terminalid,userid" を指定します。	オプション
IletPreserve	ダウンロード即起動アプリの保存不可を表すフラグ。 有効な値は"deny"（ASCII形式とし、大文字・小文字の区別はしません）のみとし、"deny"以外の文字を設定した場合は、アプリケーションの起動を禁止します。 この項目が設定されている場合は、即起動で起動されたアプリを保存しようとする例外が発生します。	オプション

LaunchApp	アプリ連携機能の使用可否を表すフラグ。 有効な値は "yes"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"yes"以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、アプリケーションの連携起動はしないものとみなし、アプリケーションの連携起動をしようとすると例外が発生します。	オプション
RemoteControl	i アプリがリモート制御アプリケーションかどうかを表します。リモート制御アプリケーションの時は、"yes"を設定します。	オプション
FeliCaCPID	FeliCa 用 コンテンツプロバイダの識別子。 FeliCaCPIDボタンを押し、ユニークなコードを付与します（図 4-18 参照）。 フリー領域用 API を使用する場合は、この項目を設定します。 32 桁の 16 進数文字列を設定してください。 ※ 仮想FeliCaデータファイルを作成しても、フリー領域の CPID Data Blockには自動的に反映されません。仮想FeliCaデータファイルビューワを利用して手動にて設定する必要があります（「6.3. 仮想FeliCaデータファイルビューワ」を参照）。	フリー領域用 API を使用する場合は必須
FeliCaPIN	FeliCa 用 対応するフリー領域に設定する PIN 情報。 フリー領域用 API を使用する場合は、この項目を設定してください。 100000 以上 8 桁以内の 10 進数文字列（100000～99999999）を設定してください。 ※ 仮想FeliCaデータファイルを作成しても、フリー領域の PIN Data Blockに自動的に反映されません。仮想FeliCaデータファイルビューワを利用して手動にて設定する必要があります（「6.3. 仮想FeliCaデータファイルビューワ」を参照）。	フリー領域用 API を使用する場合は必須
AppIcon	アプリケーションのアイコンとして使用する画像ファイル名。（最大 255 バイト） 複数の画像ファイルを設定する場合は、カンマ（,）で区切って設定します。 設定できる文字列は ASCII 形式とし、ディレクトリ階層（¥ や /）が指定された場合、255 バイトを超えて設定した場合は、アプリケーションの起動を禁止します。	オプション
UseStorage	外部メモリアクセス機能利用可否を表すフラグ。 有効な値は "ext"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"ext"以外の文字を設定した場合は、アプリケーションの起動を禁止します。	オプション

CPBindingAuth	CP の署名及び証明書。 ダウンロードしようとしている i アプリが SD-Binding 機能を利用する場合に、コンテンツプロバイダの署名と証明書を示す値を設定します。 CPBindingAuth は、FOMA のユーザ証明書でも設定することができます。FOMA のユーザ証明書でこのキーを設定するには、「6.8. CPBindingAuth 署名付与ツール」を参照してください。	オプション
CPName	CP 名を表す文字列。（最大 16 バイト） この文字列は、ストレージデバイス上に CP バインドで生成したフォルダの情報表示に利用します。 設定できる文字列は ASCII 形式とし、16 バイトを超えた文字列が設定されている場合は、各携帯端末に依存します。	オプション
GetLocationParam	ブラウザ、メーラ、トルカから起動された場合の位置情報の取得可否を表すフラグ。 有効な値は "yes"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、"yes" 以外の文字を設定した場合は、アプリケーションの起動を禁止します。 また、この項目に何も設定しない場合は、ブラウザ、メーラ、トルカから位置情報が渡りません。	オプション

このアプリケーションディスクリプタファイル（ADF）は、
「C:\iDKDoJa5.0\apps\<プロジェクト名>\bin」
に「アプリケーション名.jam」というファイルで作成されます。

4.7.1. FeliCaCPID の設定

ADF の FeliCaCPID は ADF の URL からハッシュ値を作成します。

FeliCaCPIDの設定手順を図 4-18 で説明します。

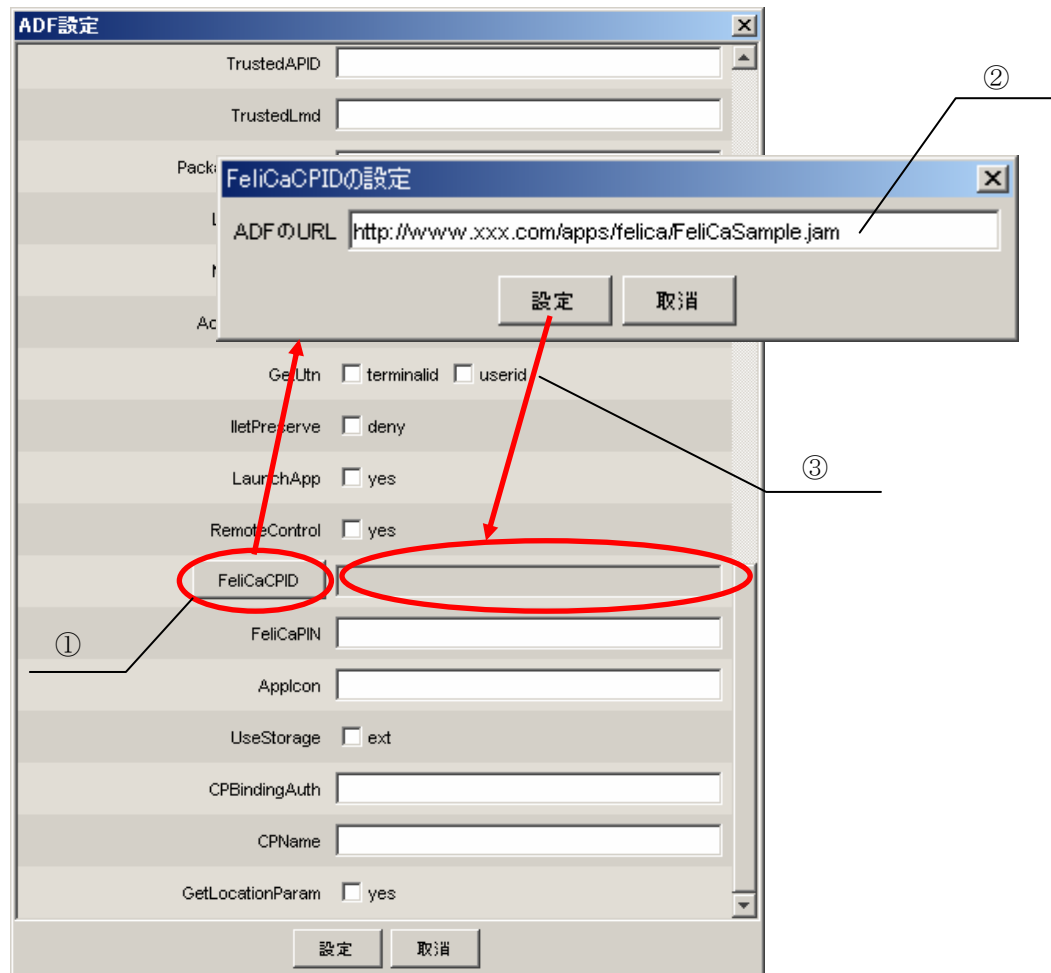


図 4-18 FeliCaCPID 設定画面

- ① ADF設定画面から「FeliCaCPID」ボタンを押すと、FeliCaCPID設定画面が表示されます（図 4-18）。
- ② FeliCaCPID 設定画面の「ADF の URL」に ADF ファイルをダウンロードする URL をスキーム(http://)から絶対パスで指定します。
- ③ FeliCaCPID 設定画面の「設定」ボタンを押します。ADF の URL をハッシュした値が ADF 設定画面に表示されます。
ADF の URL は「アプリケーション.properties」ファイルに保持され次回から ADF の URL に表示されます。
- ④ 「取消」ボタンを押した場合、FeliCaCPID は付与されず、ADF 設定画面に戻ります。

ここで付与した FeliCaCPID を削除したい場合は、ADF の URL を空にして設定してください。

4.7.2. FeliCa フリー領域の関連付け

ADF に FeliCaCPID と FeliCaPIN が設定されている場合、自動的に仮想 FeliCa のフリー領域に FeliCaCPID と FeliCaPIN を関連付ける事ができます。

A D F に FeliCaCPID と FeliCaPIN が設定されていると A D F 設定画面の「設定」を押したタイミングでどのフリー領域に関連付けるかを問い合わせます（図 4-19）。

一度、問い合わせると変更しない限り 2 回目以降は問い合わせません。

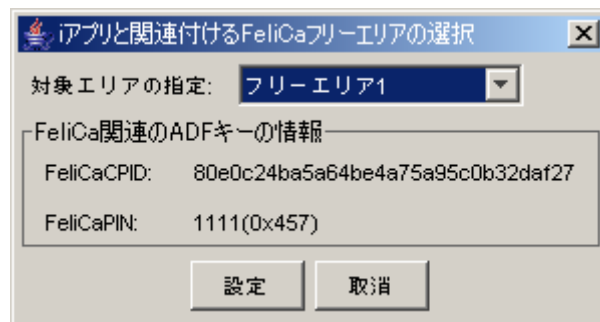


図 4-19 フリーエリアの選択

対象となるフリー領域を選択して「設定」を押すと、そのフリー領域に関連付けられます。

4.7.3. LastModified 項目の表記方法

ADF の LastModified 項目の表記方法を下記のとおりに規定します。

- 表記法は、RFC2822 の「3.3. Date and Time Specification」に準拠します。ただし、「4.3. Obsolete Date and Time」に記述されている内容には対応しておりません。以下に規定した項目については、これに従うものとします。
- 日付の表記は、ゼロサプレスの有無に関わらず可能とします。
例:
Sun, 1 Apr 2001 19:05:50 +0900 /* OK (ゼロサプレス無) */
Sun, 01 Apr 2001 19:05:50 +0900 /* OK (ゼロサプレス有) */
- 時、分、秒の表記は、ゼロサプレスの有無に関わらず可能とします。
例:
Sun, 01 Apr 2001 19: 5:50 +0900 /* OK (ゼロサプレス無) */
Sun, 01 Apr 2001 19:05:50 +0900 /* OK (ゼロサプレス有) */
- 曜日名の表記は、“Mon”、“Tue”、“Wed”、“Thu”、“Fri”、“Sat”、“Sun”とします。大文字・小文字の区別はしません。これに合致しない場合は、不正とみなします。
例: MON, mon, MoN, mOn /* OK */
- 月名の表記は、“Jan”、“Feb”、“Mar”、“Apr”、“May”、“Jun”、“Jul”、“Aug”、“Sep”、“Oct”、“Nov”、“Dec”とします。大文字・小文字の区別はしません。これに合致しない場合は、不正とみなします。
例: JAN, jan, JaN, jAn /* OK */
- 日付と時刻の有効範囲を閏年まで含めてチェックします。但し、日付と曜日の整合性はチェックしません。
例:
Sun, 01 Apr 2001 19:05:50 +0900 /* OK */
Tue, 01 Apr 2001 19:05:50 +0900 /* OK (曜日の整合性チェックなし) */
Sun, 01 Apr 2001 25:05:50 +0900 /* NG (時刻が不正) */
Sun, 31 Apr 2001 19:05:50 +0900 /* NG (4/31 は存在しない) */
Sun, 29 Feb 2001 19:05:50 +0900 /* NG (2001/2/29 は存在しない) */
Sun, 29 Feb 2004 19:05:50 +0900 /* OK (2004/2/29 は存在する) */
- タイムゾーンの表記は省略可能とし、省略された場合は+0900 と解釈します。また、タイムゾーンのアルファベット表記 (“GMT”、“JST”) も可能とします。

4.8. 待ち受け起動

「待ち受け起動」とは、i アプリを「待ち受けアプリケーション」として起動することです。「待ち受け起動」の起動タイプは、「LAUNCHED_AS_CONCIERGE」です。

4.8.1. 待ち受けアプリケーション

「待ち受けアプリケーション」とは、待ち受け画面になることができる i アプリのことで、com.nttdocomo.ui.MApplication クラスを継承している必要があります。表 4-4 に「待ち受け画面」と「待ち受けアプリケーション」との比較を示します。

表 4-4 「待ち受け画面」と「待ち受けアプリケーション」の比較

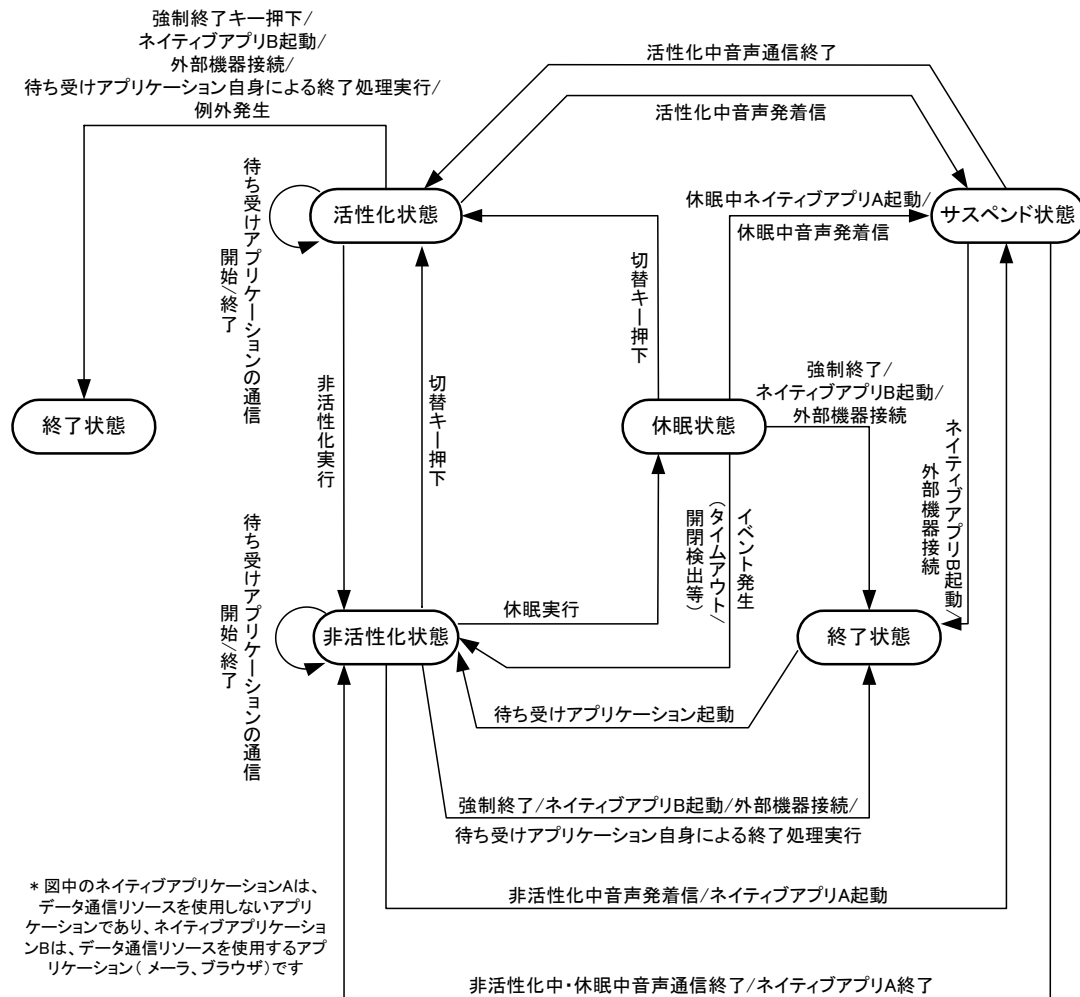
待ち受け画面設定	表示	動作
待ち受け画像 (GIF・JPEG)	静止画またはアニメ画像	キー押下、開閉に合わせて アニメ表示
待ち受けアプリケーション	待ち受けアプリケーションに よる描画	特定キー押下でキーイベン トを有効化

待ち受けアプリケーションは、通常のアプリケーションと同様にキー入力を受け取る状態（**活性化状態**）と、待ち受け画面と同様に画面の表示は行うがキー入力は受け取らない状態（**非活性化状態**）及び、一時停止した状態（**休眠状態**あるいはサスペンド状態）の3つの状態を持っています。

待ち受けアプリケーションが実行中であっても、非活性化状態 及び 休眠状態 では端末のネイティブ機能を優先します。

待ち受けアプリケーションが 非活性化状態 及び 休眠状態 では、キーイベントの受信とソフトキーの表示は行えず、全てネイティブアプリケーション側で処理します。一方、活性化状態ではキーイベントの受信とソフトキーの表示は、待ち受けアプリケーションで行います。

図 4-20 に待ち受けアプリケーションの状態遷移を示します。



切替キー：待ち受けボタン

図 4-20 待ち受けアプリケーションの状態遷移

待ち受けアプリケーションのプログラムは、「プロジェクト」→「ソースファイルの新規作成...」で、テンプレートを「待ち受けアプリケーション」にするだけで簡単に作成する事ができます（「4.2.1. ソースファイルの作成」参照）。

4.8.2. 待ち受け起動手順

エミュレータで待ち受けアプリケーションを起動するには：

1. メイン画面のプルダウンから「待ち受け起動」を選択します（図 4-2 1）。

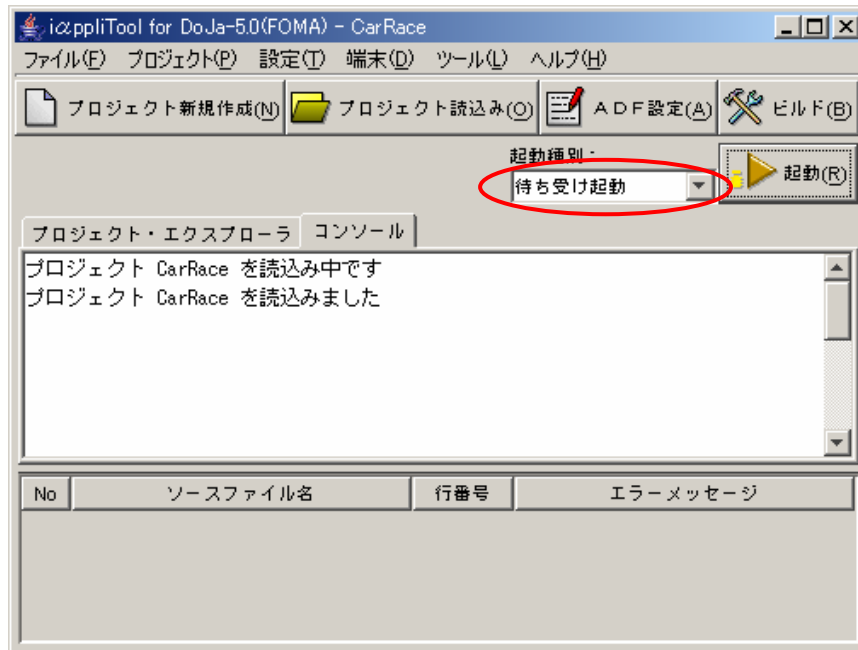


図 4-2 1 待ち受け起動準備

2. メニューから「プロジェクト」→「起動...」を選択するか、「起動」ボタンを押します。
i アプリを待ち受けアプリケーションとして起動するためには、以下の条件を満たしている必要があります：

- i アプリを待ち受けアプリケーションとして作成していること（com.nttdocomo.ui.MApplication クラスを継承していること）。
- ADFの項目で「MyConcierge」に「Yes」が設定されていること（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」を参照）。
- メーラ（「4.1 6. メーラ起動」を参照）、ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動していないこと。

com.nttdocomo.ui.MApplicationクラスの継承の有無とMyConciergeの設定の有無との関係を表 4-5 に示します。

表 4-5 MApplication クラス継承の有無と MyConcierge 設定の有無の関係

	MyConcierge 設定有	MyConcierge 設定無
MApplication 継承有	起動	エラー
MApplication 継承無	エラー	エラー

上記の条件を満たしていない場合には、i アプリの起動を禁止するダイアログあるいはコンソールにエラーメッセージが表示されます。

上記の条件を満たしている場合には、i アプリを待ち受けアプリケーションとして起動します（図 4-2 2）。



図 4-2 2 待ち受けアプリケーション起動

3. 待ち受けアプリケーションが「非活性化状態」あるいは「休眠状態」にあるときには、以下のアプリケーションが起動可能です：
 - ・ 「待ち受け起動」以外の別の i アプリ
 - ・ メーラ（「4.1 6. メーラ起動」を参照）
 - ・ ブラウザ（「4.1 7. ブラウザ起動」を参照）
 - ・ コードリーダー（「4.1 8. コードリーダー起動」を参照）
 - ・ トルカ（「4.2 3. トルカ起動」を参照）
 - ・ BML ブラウザ（「4.2 5. BML 起動」を参照）
 - ・ DTV（「4.2 6. DTV 起動」を参照）

また、上記のアプリケーションが終了すると自動的に待ち受けアプリケーションが「非活性化状態」で再起動されます。

以下（図 4-2 3）は、待ち受けアプリケーションのソースコードのサンプルです。
 （このソースコードの場合は、
C:\iDKDoJa5.0\apps\<プロジェクト名>\src\MAppli.java を作成します。）

```

/**
 * 待ち受けアプリケーション
 */
import com.nttdocomo.ui.*;

public class MAppli extends MApplication implements SoftKeyListener {
    Panel        p;
    MediaImage    image;
    ImageLabel    imageLb;
    Label         lb;
    boolean       flag = false;

    public void start() {
        try {
            // イメージとラベルの表示
            p = new Panel();
            lb = new Label("[待受] を押してください。");
            imageLb = new ImageLabel();
            image = MediaManager.getImage("resource:///sample1.gif");
            image.use();
            imageLb.setImage(image.getImage());
            HTMLLayout lm = new HTMLLayout();
            p.setLayoutManager(lm);
            lm.begin(HTMLLayout.CENTER);
            p.add(imageLb);
            lm.br();
            p.add(lb);
            lm.end();
            p.setSoftKeyListener(this);
        } catch (Exception e) {
        }
        Display.setCurrent(p);
    }

    /**
     * システムイベント
     */
    public void processSystemEvent(int type, int param) {
        if (type == MODE_CHANGED_EVENT) {
            p.setSoftLabel(Frame.SOFT_KEY_1, "戻る");
            lb.setText("活性化状態");
            flag = false;
        }
        // フリップ変更イベント
        else if (type == FOLD_CHANGED_EVENT) {
            if (param == 0) { // 閉じる
                if (flag) {
                    lb.setText("休眠状態");
                    sleep(); // 休眠状態へ（省電力モードにする）
                    lb.setText("非活性化状態");
                }
            }
        }
    }

    /**
     * ソフトキーイベント
     */
    public void softKeyPressed(int no) {
    }
    public void softKeyReleased(int no) {
        // ソフトキー 1 押下
        if (no == Frame.SOFT_KEY_1) {
            p.setSoftLabel(Frame.SOFT_KEY_1, "");
            lb.setText("非活性化状態");
            deactivate(); // 非活性化状態
            flag = true;
        }
    }
}

```

図 4-2 3 （例）待ち受けアプリケーションソースコード

（lib\Multimedia\sample1.gif を apps\<プロジェクト名>\res にコピーします。）

4.9. 受信フォルダ起動

受信フォルダ起動とは、i アプリをメールの受信フォルダから起動する方法で「通常起動」とは以下の点が異なります：

- トラストド i アプリです（「5.2. トラストド動作の設定」参照）。
- 「受信フォルダ起動」の起動タイプは「LAUNCHED_MSG_RECEIVED」です。
- ADFの設定項目「MessageCode」、「TrustedAPID」に適切な値を設定し、「トラストド動作設定」→「メッセージ i アプリ機能の許可」を yes にした場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

実際の端末において、「通常起動」とは通常のメニューからの起動であり、「受信フォルダ起動」とは受信メールボックスのメールから起動した場合に相当するものです。

受信フォルダ起動するには、プロジェクトを読み込み、ビルド、ADFの設定を行った後、図 4-2 4 に示すように、メイン画面のプルダウンから「受信フォルダ起動」を選択し、「起動」ボタンを押します。

また、起動の際の制約などに関しては「通常起動」の場合と同様です（「4.3. プロジェクトのビルドと通常起動」参照）。

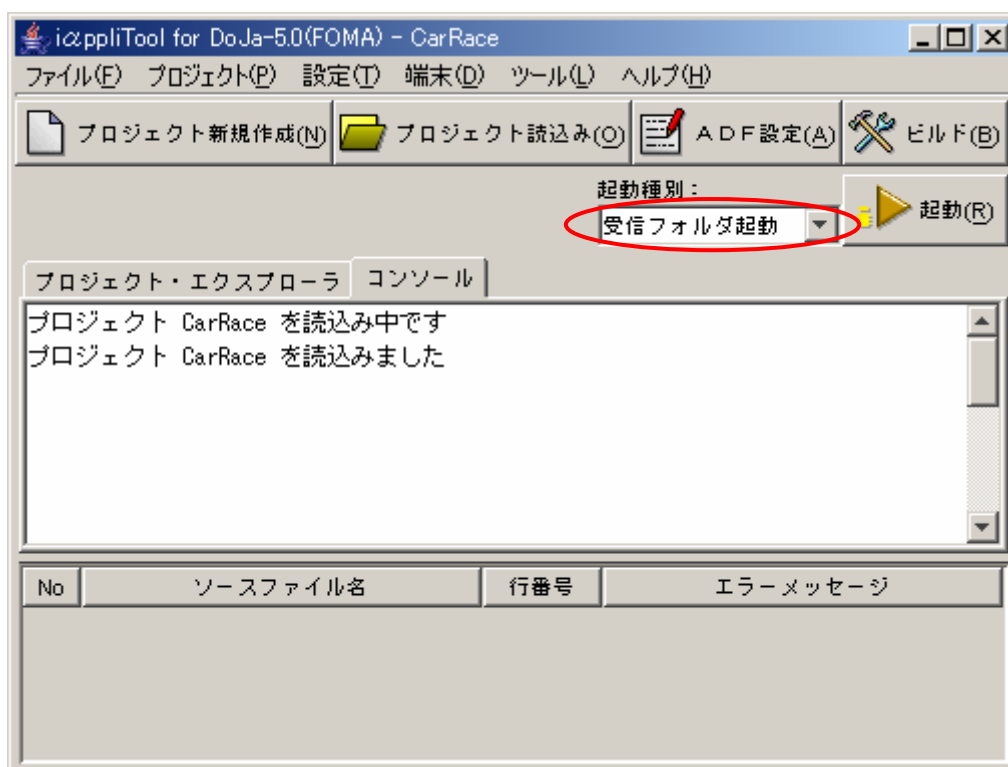


図 4-2 4 受信フォルダ起動準備

4.10. 送信フォルダ起動

送信フォルダ起動とは、i アプリをメールの送信フォルダから起動する方法で「通常起動」とは以下の点が異なります：

- トラステッドi アプリです（「5.2. トラステッド動作の設定」参照）。
- 「送信フォルダ起動」の起動タイプは「LAUNCHED_MSG_SENT」です。
- ADFの設定項目「MessageCode」、「TrustedAPID」に適切な値を設定し、「トラステッド動作設定」→「メッセージi アプリ機能の許可」を yes にした場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

実際の端末において、「通常起動」とは通常のメニューからの起動であり、「送信フォルダ起動」とは送信メールボックスのメールから起動した場合に相当するものです。

送信フォルダ起動するには、プロジェクトを読み込み、ビルド、ADFの設定を行った後、図 4-25 に示すように、メイン画面のプルダウンから「送信フォルダ起動」を選択し、「起動」ボタンを押します。

また、起動の際の制約などに関しては「通常起動」の場合と同様です（「4.3. プロジェクトのビルドと通常起動」参照）。

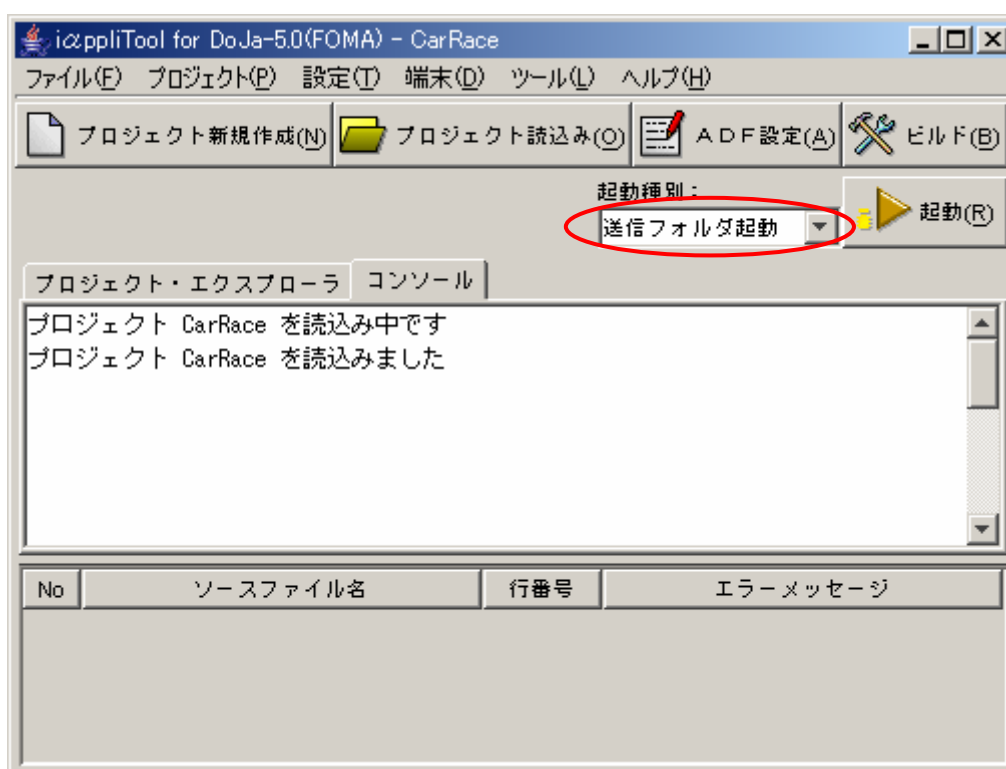


図 4-25 送信フォルダ起動準備

4.11. 未送信フォルダ起動

未送信フォルダ起動とは、i アプリをメールの未送信フォルダから起動するで「通常起動」とは以下の点が異なります：

- トラストedd i アプリです（「5.2. トラストedd動作の設定」参照）。
- 「未送信フォルダ起動」の起動タイプは「LAUNCHED_MSG_UNSENT」です。
- ADFの設定項目「MessageCode」、「TrustedAPID」に適切な値を設定し、「トラストedd動作設定」→「メッセージ i アプリ機能の許可」を yes にした場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

実際の端末において、「通常起動」とは通常のメニューからの起動であり、「未送信フォルダ起動」とは未送信メールボックスのメールから起動した場合に相当するものです。

未送信フォルダ起動するには、プロジェクトを読み込み、ビルド、ADFの設定を行った後、図 4-26 に示すように、メイン画面のプルダウンから「未送信フォルダ起動」を選択し、「起動」ボタンを押します。

また、起動の際の制約などに関しては「通常起動」の場合と同様です（「4.3. プロジェクトのビルドと通常起動」参照）。

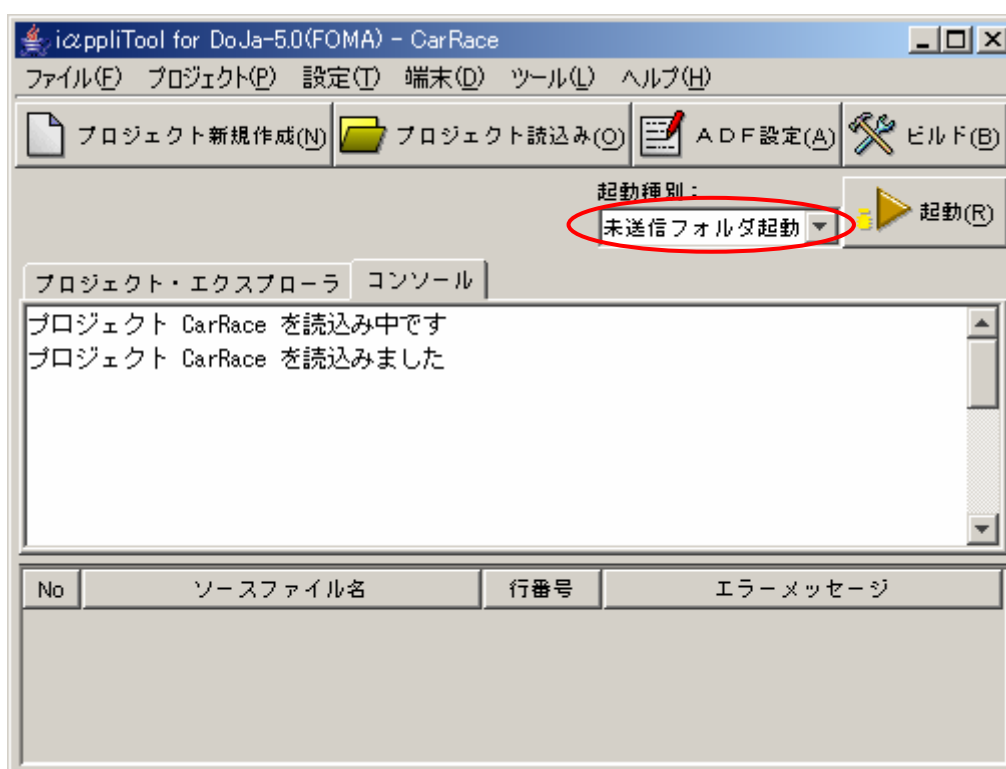


図 4-26 未送信フォルダ起動準備

4.12. 初回起動

初回起動とは、i アプリを始めて起動する方法で「通常起動」とは以下の点異なります：

- 「初回起動」の起動タイプは「LAUNCHED_AFTER_DOWNLOAD」です。

実際の端末において、「通常起動」とは通常のメニューからの起動であり、「初回起動」とはダウンロード直後（通常起動の1回目）の起動に相当するものです。

初回起動するには、プロジェクトを読み込み、ビルド、ADFの設定を行った後、図 4-27 に示すように、メイン画面のプルダウンから「初回起動」を選択し、「起動」ボタンを押します。

また、起動の際の制約などに関しては「通常起動」の場合と同様です（「4.3. プロジェクトのビルドと通常起動」参照）。

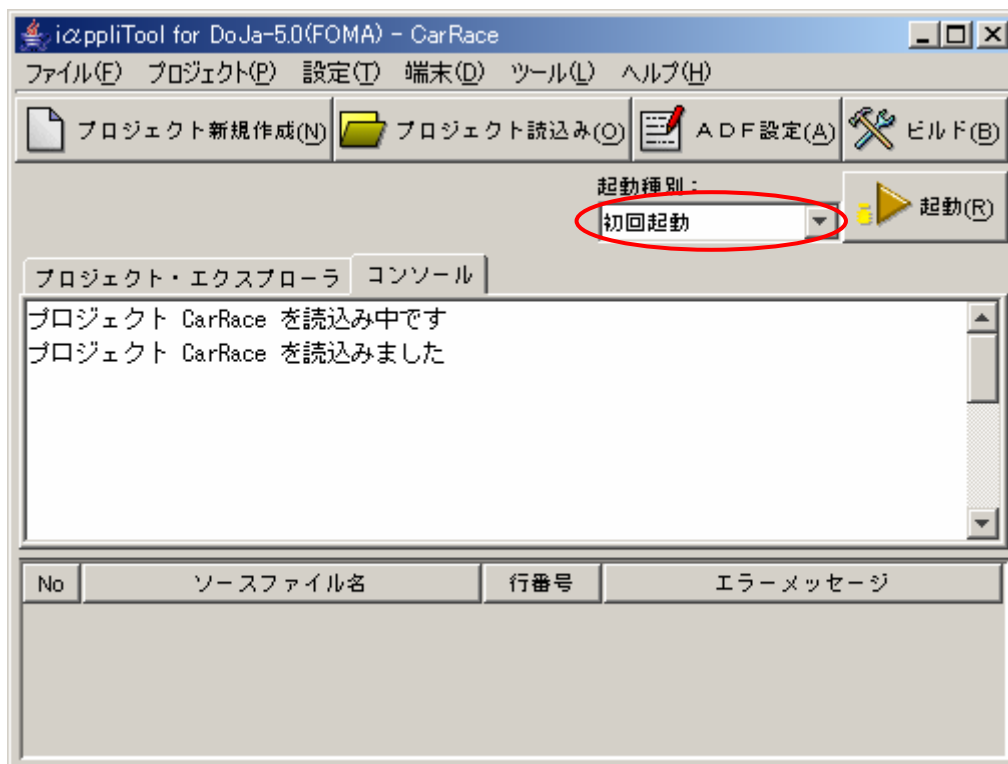


図 4-27 初回起動準備

4.13. タイマ起動

タイマ起動とは、i アプリをタイマによって起動する方法で「通常起動」とは以下の点が異なります：

- 「タイマ起動」の起動タイプは「LAUNCHED_FROM_TIMER」です。
- ADFの設定項目「LaunchAt」に適切な値を設定した場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

実際の端末において、「通常起動」とは通常のメニューからの起動であり、「タイマ起動」とはタイマエージェントにより指定された時刻の起動に相当するものです。

タイマ起動するには、プロジェクトを読み込み、ビルド、ADFの設定を行った後、図 4-28 に示すように、メイン画面のプルダウンから「タイマ起動」を選択し、「起動」ボタンを押します。

また、起動の際の制約などに関しては「通常起動」の場合と同様です（「4.3. プロジェクトのビルドと通常起動」参照）。

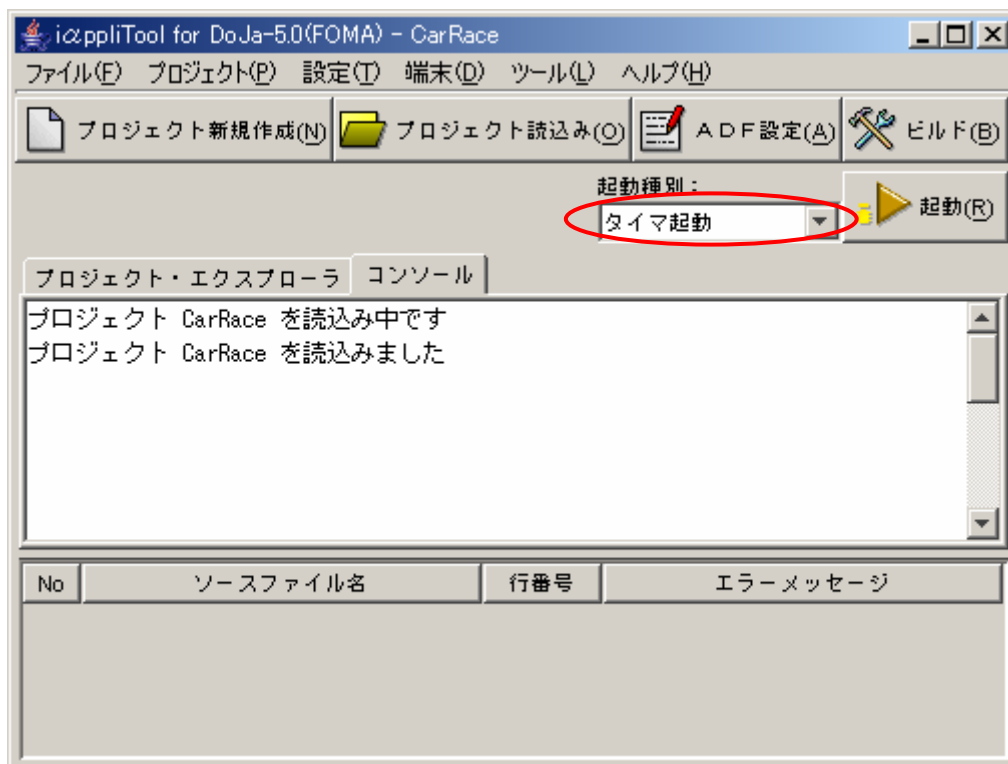


図 4-28 タイマ起動準備

4.14. 外部機器起動

外部機器起動とは、i アプリを外部機器から起動する方法で「通常起動」とは以下の点が異なります：

- 「外部機器起動」の起動タイプは「LAUNCHED_FROM_EXT」です。
- ADFの設定項目「AllowPushBy」に適切な値を設定した場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

上記の条件であった場合には、例外が発生します。

実際の端末において、「通常起動」とは通常のメニューからの起動であり、「外部機器起動」とは外部インタフェースのトリガによる起動に相当するものです。

4.14.1. 外部機器起動手順

1. メイン画面のプルダウンから「外部機器起動」を選択します（図 4-2 9）。

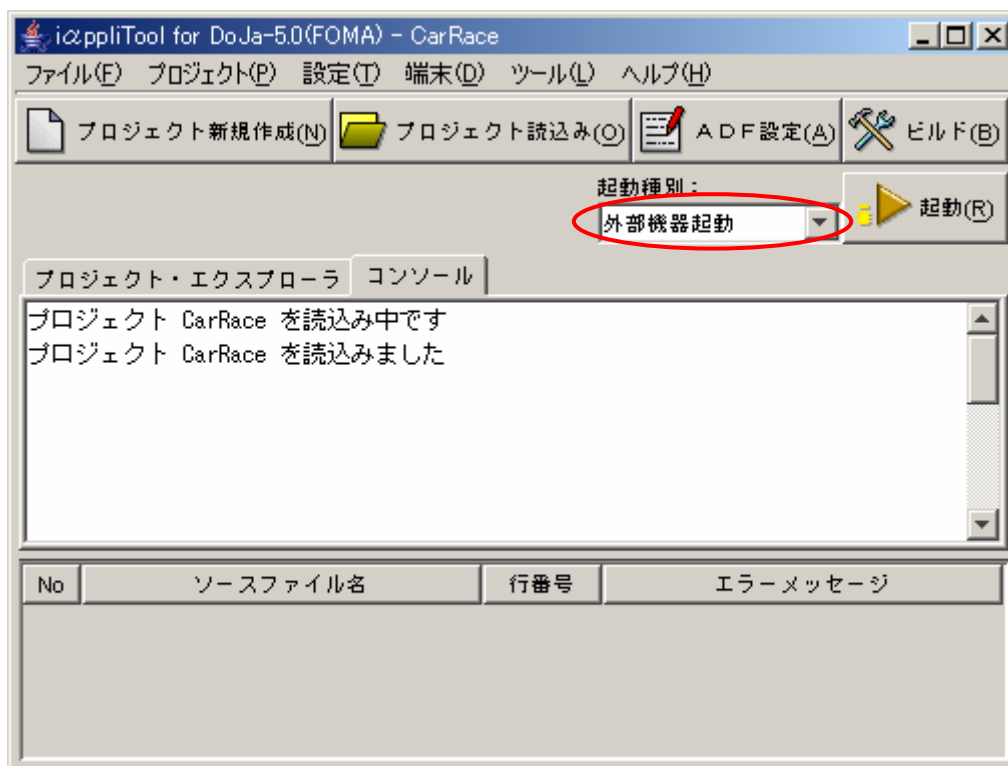


図 4-2 9 外部機器起動準備

2. 起動パラメータの入力画面が表示されるので、起動パラメータを以下のように入力します（図 4-30）。

パラメータキー1 パラメータ値1 パラメータキー2 パラメータ値2・・・

※ 起動パラメータは16個まで指定可能です。

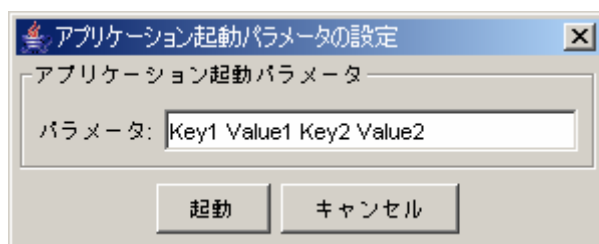


図 4-30 起動パラメータの入力画面

3. 「起動」ボタンを押します。

4.15. 即起動

「即起動」とは、ダウンロード後に i アプリを即起動することです。

「即起動」の起動タイプは、「LAUNCHED_AS_ILET」です。

「即起動」で起動された i アプリは、以下の機能を利用することができません：

- A D F の"GetSysInfo" で許可される機能
- A D F の"GetUtn" で許可される機能
- A D F の"AccessUserInfo" で許可される機能
- A D F の"LaunchApp" で許可される機能
- A D F の"TrustedAPID" が指定される機能
(トラステッド動作設定で "Yes" に設定する機能)
- スクラッチパッドの利用
- i アプリのバージョンアップ
- i アプリの保存

上記の条件であった場合には、例外が発生します。

4.15.1. 即起動手順

1. メイン画面のプルダウンから「即起動」を選択します（図 4-3 1）。

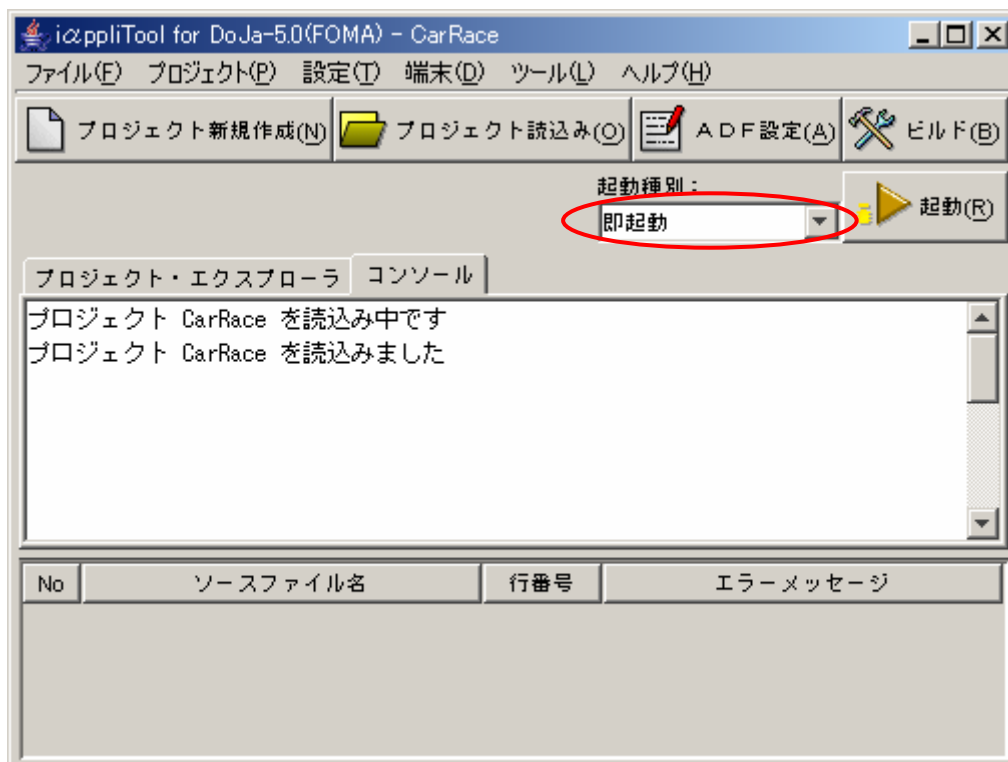


図 4-3 1 即起動準備

2. 「起動」ボタンを押します。

4.16. メーラ起動

「メーラ起動」とは、i アプリをメーラから起動することです。

「メーラ起動」の起動タイプは、「LAUNCHED_FROM_MAILER」です。

4.16.1. メーラ起動手順

エミュレータで i アプリをメーラから起動するには：

1. メイン画面のプルダウンから「メーラ起動」を選択します（図 4-3 2）。

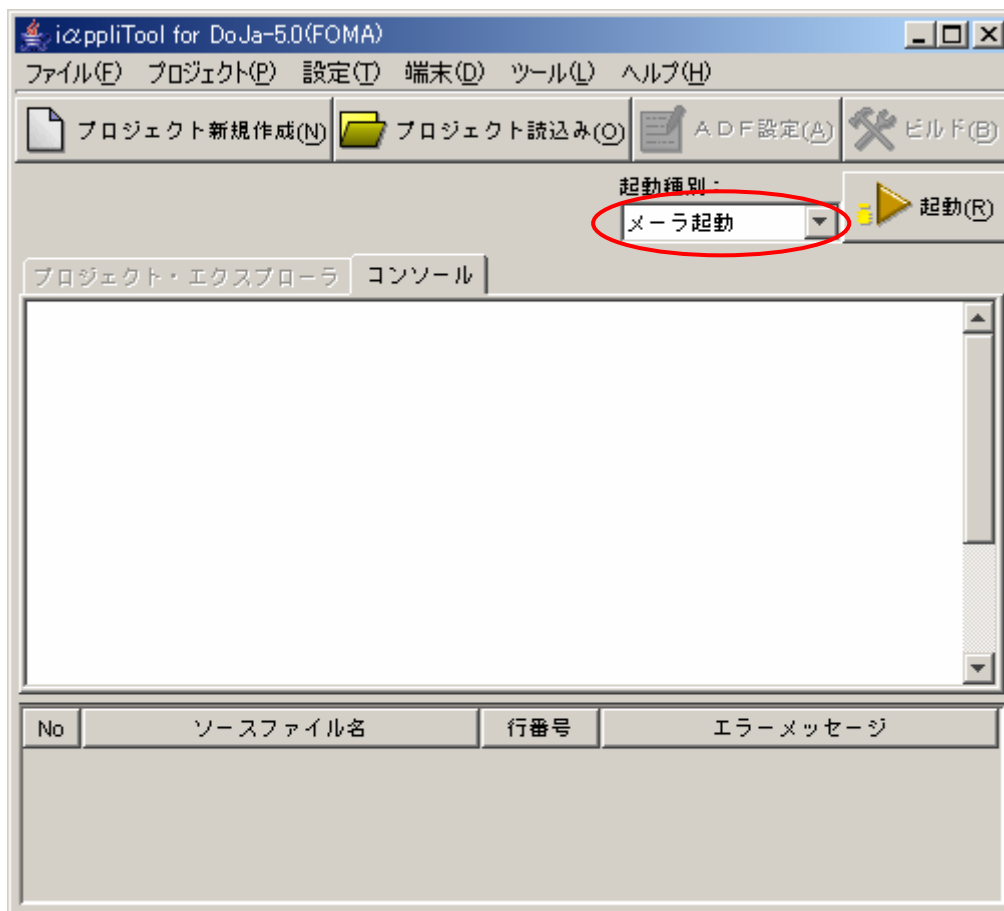


図 4-3 2 メーラ起動準備

2. メーラ起動ファイルを編集して起動するアプリを追記します（「10.1. メーラ起動ファイル」を参照）。

3. 「起動」ボタンを押します。

メーラ画面が表示されます（図 4-3 3）。このとき、エミュレータが以下の状態の時にはメーラ画面は表示されません:

- 既にメーラが起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。
- コードリーダー（「4.1 8. コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

メーラ画面の上方には、受信メールの一覧が表示されます（メーラ起動ファイル *.mail（*は0、1、2、・・・）のヘッダ部分が表示されます）。

受信メールの一覧からメールを選択すると、メーラ画面の下方に選択したメールの内容が表示されます。

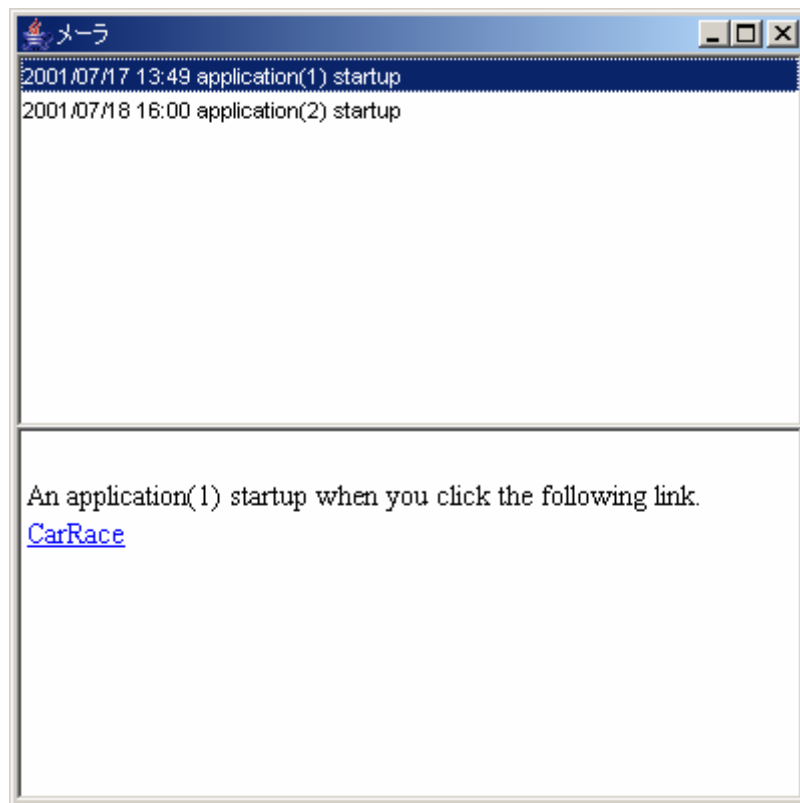


図 4-3 3 メーラ画面

4. メーラ画面の下方に表示されているリンク（青い文字）をクリックします。

i アプリをメーラから起動するためには、以下の条件を満たしている必要があります：

- ADFの項目で「LaunchByMail」に「any」あるいは起動を許可するメールアドレスを設定し（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」を参照）、そのメールアドレスがメールのFromと一致していること。
- メーラ起動ファイルに起動する i アプリが正しく記述されていること（「10.1. メーラ起動ファイル」を参照）。
- i アプリが「プロジェクト読み込み画面」で指定したプロジェクト格納先に作成されていること。必ずしもデフォルトの格納先（「C:\iDKDoJa5.0\apps」）に作成されている必要はありません。

上記の条件を満たしていない場合には、i アプリの起動を禁止するダイアログあるいはコンソールにエラーメッセージが表示されます。

上記の条件を満たしている場合には、i アプリをメーラ起動するための確認ダイアログが表示され（図 4-3 4）、「はい」ボタンを押すと i アプリが起動します（図 4-3 5）。

メーラから待ち受けアプリケーションを起動した場合、待ち受けアプリケーションとしては動作せず通常のアプリケーションとして動作します。

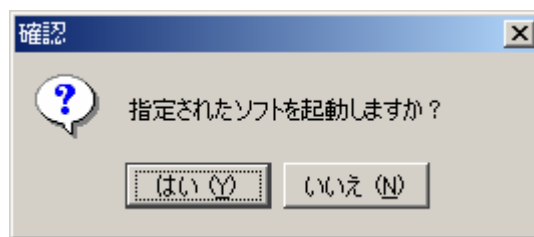


図 4-3 4 起動確認ダイアログ



図 4-3 5 メーラ起動された i アプリ

5. メーラ起動された i アプリの終了方法により、その後、自動起動する i アプリが異なります:
 - メーラ起動された i アプリを正常終了 (i アプリが `IApplication#terminate()` メソッドを呼び出して終了した場合)

再びメーラ画面が表示されます。メーラ画面を終了する (画面の右上の「×」を押す) と、もし、メーラを起動する際に待ち受けアプリケーションが動作していたのであればそれを起動し、そうでなければ通常の待ち受け画面に戻ります。

- メーラ起動された i アプリを強制終了（端末の「終話ボタン」を押した場合）

メーラ画面は表示されず、もし、メーラを起動する際に待ち受けアプリケーションが動作していたのであればそれを起動し、そうでなければ通常の待ち受け画面に戻ります。

4.17. ブラウザ起動

「ブラウザ起動」とは、i アプリをブラウザから起動することです。
「ブラウザ起動」の起動タイプは、「LAUNCHED_FROM_BROWSER」です。

4.17.1. ブラウザ起動手順

エミュレータで i アプリをブラウザから起動するには：

1. メイン画面のプルダウンから「ブラウザ起動」を選択します（図 4-3 6）。

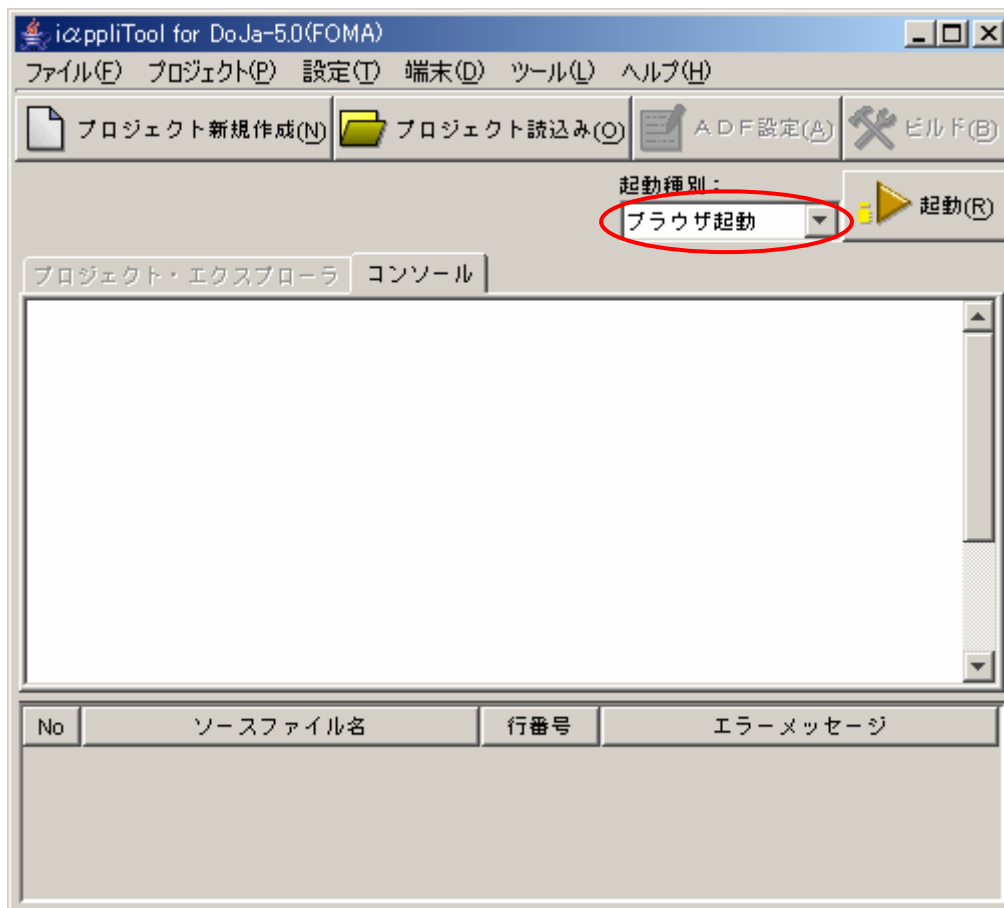


図 4-3 6 ブラウザ起動準備

2. 「起動」ボタンを押します。

ブラウザ画面が表示されます（図 4-3 7）。このとき、エミュレータが以下の状態の時にはブラウザ画面は表示されません：

- 既にブラウザが起動している。
- メーラ（「4.1 6 メーラ起動」を参照）が起動している。
- コードリーダー（「4.1 8 コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3.トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

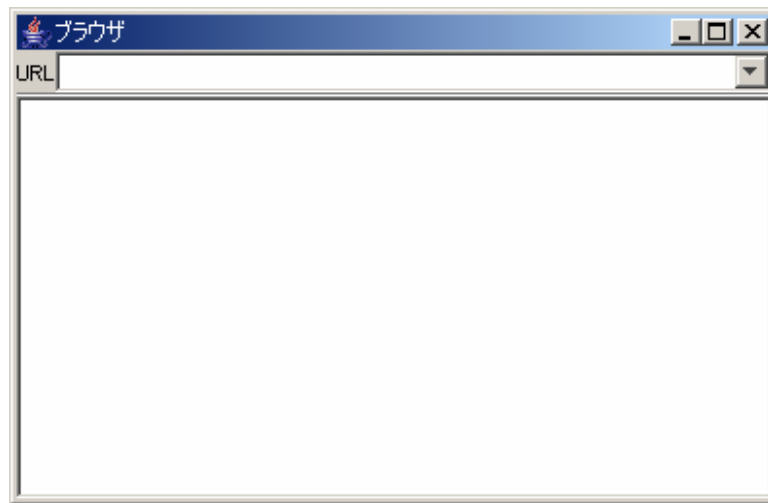


図 4-3 7 ブラウザ画面

3. ブラウザに表示するHTMLファイルを編集して起動するアプリを追記します（「10.4. ブラウザ起動HTMLファイルの記述方法」参照）。
4. 図 4-3 7 の上方のURLフィールドに 3. で編集したHTMLのURLを入力し、リターンキーを押します（図 4-3 8）。



図 4-3 8 URL の表示

- 図の URL は、同一マシン(localhost)に Web サーバ（Apache 等）を導入し、
C:/iDKDoJa5.0/lib/sample.html を用いた場合です。
もし、Web サーバを利用しない場合は、「file:」で指定できます。
その場合は、URL に「**file:C:/iDKDoJa5.0/lib/sample.html**」と指定
します。

5. 図 4-38 のリンク（青い文字）をクリックします。

i アプリをブラウザから起動するためには、以下の条件を満たしている必要があります：

- ADFの項目で「LaunchByBrowser」に「any」あるいは起動を許可するURLが
設定されていること（「4.7.アプリケーションディスクリプタファイル
（ADF）の編集」を参照）。
- HTMLファイルに起動するアプリが正しく指定されていること（「10.4.
ブラウザ起動HTMLファイルの記述方法」参照）。
- i アプリが「プロジェクト読み込み画面」で指定したプロジェクト格納先に作
成されていること。必ずしもデフォルトの格納先（「C:/iDKDoJa5.0/apps」）
に作成されている必要はありません。

上記の条件を満たしていない場合には、i アプリの起動を禁止するダイアログあるいは
コンソールにエラーメッセージが表示されます。

上記の条件を満たしている場合には、i アプリをブラウザ起動するための確認ダイア
ログが表示され（図 4-39）、「はい」ボタンを押すと i アプリが起動します（図
4-40）。

ブラウザから待ち受けアプリケーションを起動した場合、待ち受けアプリケーション
としては動作せずに通常のアプリケーションとして動作します。

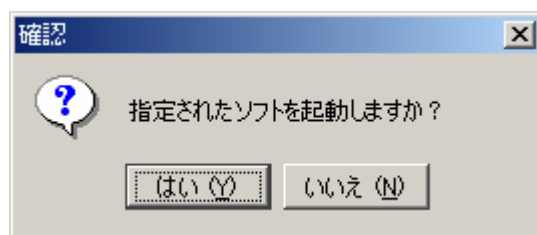


図 4-39 起動確認ダイアログ



図 4-4 0 ブラウザ起動された i アプリ

6. ブラウザ起動された i アプリの終了方法により、その後、自動起動する i アプリが異なります:
 - ブラウザ起動された i アプリを正常終了 (i アプリが `IApplication#terminate()` メソッドを呼び出して終了した場合)

再びブラウザ画面が表示されます。ブラウザ画面を終了する (画面の右上の「×」を押す) と、もし、ブラウザを起動する際に待ち受けアプリケーションが動作していたのであればそれを起動し、そうでなければ通常の待ち受け画面に戻ります。

- ブラウザ起動された i アプリを強制終了（端末の「終話ボタン」を押した場合）

ブラウザ画面は表示されず、もし、ブラウザを起動する際に待ち受けアプリケーションが動作していたのであればそれを起動し、そうでなければ通常の待ち受け画面に戻ります。

4.18. コードリーダー起動

「コードリーダー起動」とは、i アプリをコードリーダーから起動することです。
「コードリーダー起動」の起動タイプは、「LAUNCHED_FROM_EXT」です。

4.18.1. コードリーダー起動手順

エミュレータで i アプリをコードリーダーから起動するには：

1. メイン画面のプルダウンから「コードリーダー起動」を選択します（図 4-4 1）。

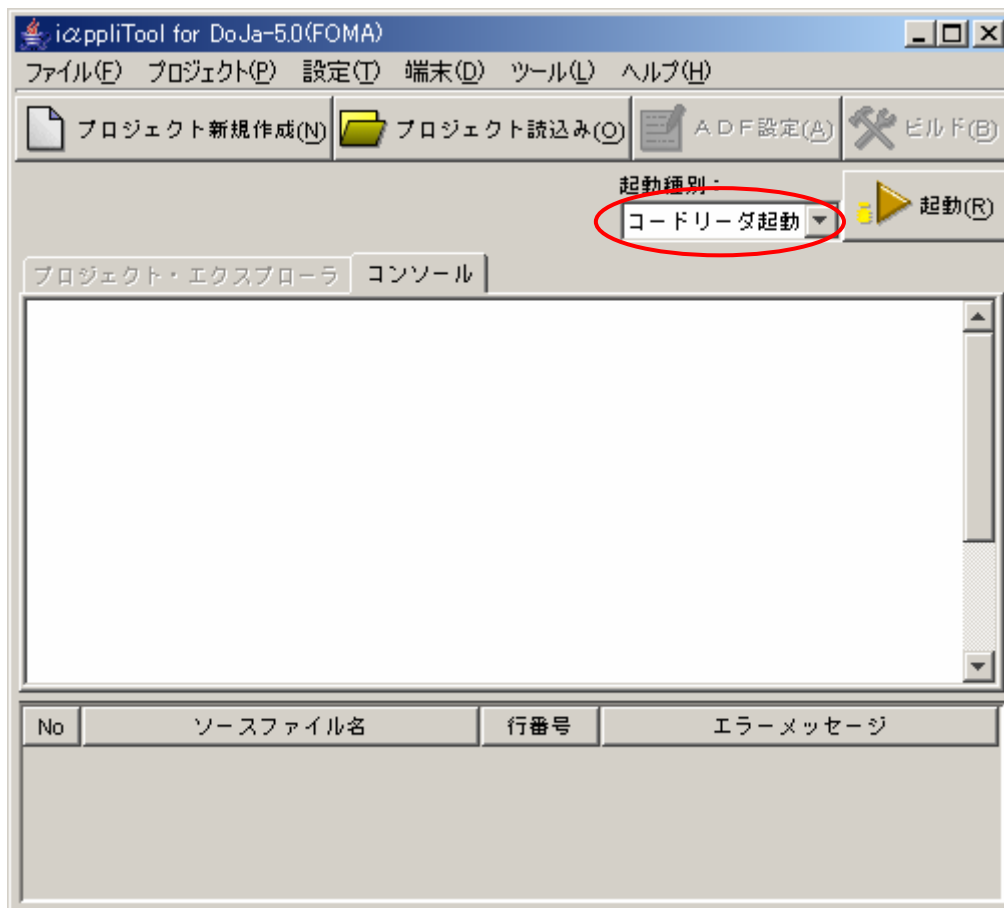


図 4-4 1 コードリーダー起動準備

2. コード認識結果ファイルを編集して起動するアプリを追記します（「10.12. コード認識結果ファイル」参照）。

3. 「起動」 ボタンを押します。

コード認識結果ファイル読み込み画面が表示されます（図 4-4 2）。
このとき、エミュレータが以下の状態の時にはコード認識結果ファイル読み込み画面は表示されません：

- 既にコードリーダーが起動している。
- メーラ（「4.1 6. メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。
- トルカ（「4.2 3. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

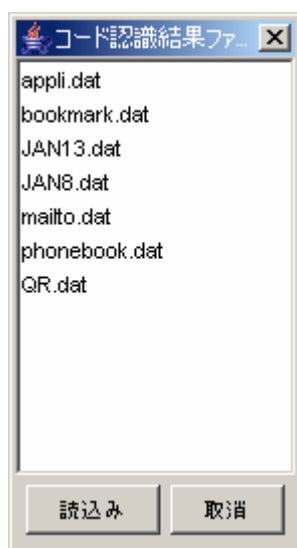


図 4-4 2 コード認識結果ファイル読み込み画面

4. 「図 4-4 2」 から認識したいコード認識ファイルを選択して「読み込み」 ボタンを押します。

i アプリをコードリーダーから起動するためには、以下の条件を満たしている必要があります：

- ADFの項目で「AllowPushBy」に「Code:起動コマンド」が設定されていること（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。
- コード認識結果ファイルのCMD:に起動する i アプリが正しく記述されていること（「1 0.1 2. コード認識結果ファイル」参照）。
- i アプリが「プロジェクト読み込み画面」で指定したプロジェクト格納先に作成されていること。必ずしもデフォルトの格納先（「C:\iDKDoJa5.0\apps」）に作成されている必要はありません。

上記の条件を満たしていない場合には、i アプリの起動を禁止するダイアログあるいはコンソールにエラーメッセージが表示されます。

上記の条件を満たしている場合には、コード認識表示画面が表示されます（図 4-4 3）。

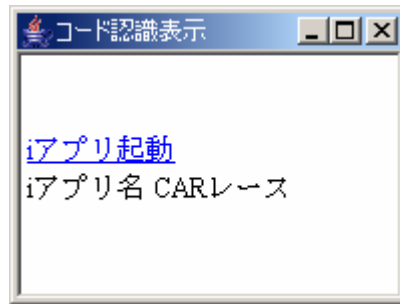


図 4-4 3 コード認識表示画面

5. 図 4-4 3 のリンク（青い文字）をクリックします。

i アプリをコードリーダーから起動するための確認ダイアログが表示され「はい」ボタンを押すと i アプリが起動します（図 4-4 4）。



図 4-4 4 コードリーダー起動された i アプリ

コードリーダーから待ち受けアプリケーションを起動した場合、待ち受けアプリケーションとしては動作せずに通常のアプリケーションとして動作します。

4.18.2. CodeReader の実行手順

「i alpha Development Kit for DoJa-5.0(FOMA)」で、アプリケーションから CodeReader#read() を実行した場合は、次の手順に従います。

1. CodeReader#read() を実行すると、コード認識結果ファイルを指定するダイアログが表示されます（図 4-4 5）。

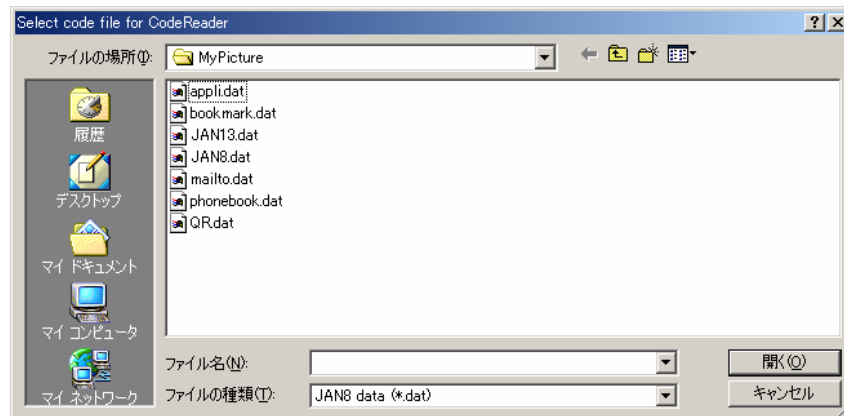


図 4-4 5 コード認識結果ファイル 指定ダイアログ

2. 読み込むコード認識結果ファイルを選択し、「ファイルの種類」から読み込むデータに該当する種類を選択して、「開く」を押します。
3. 読み込んだデータやコード種別が CodeReader に返ります。

4.19. 位置情報起動

「位置情報起動」とは、i アプリをGPSで測位した位置情報から起動する方法で「通常起動」とは以下の点が異なります：

- トラストド i アプリです（「5.2. トラストド動作の設定」参照）。
- 「位置情報起動」の起動タイプは「LAUNCHED_FROM_LOCATION_INFO」です。
- ADFの設定項目「TrustedAPID」に適切な値を設定し、「トラストド動作設定」→「位置情報取得APIの許可」を yes にした場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

4.19.1. 位置情報起動手順

1. メイン画面のプルダウンメニューから「位置情報起動」を選択します（図 4-46）。

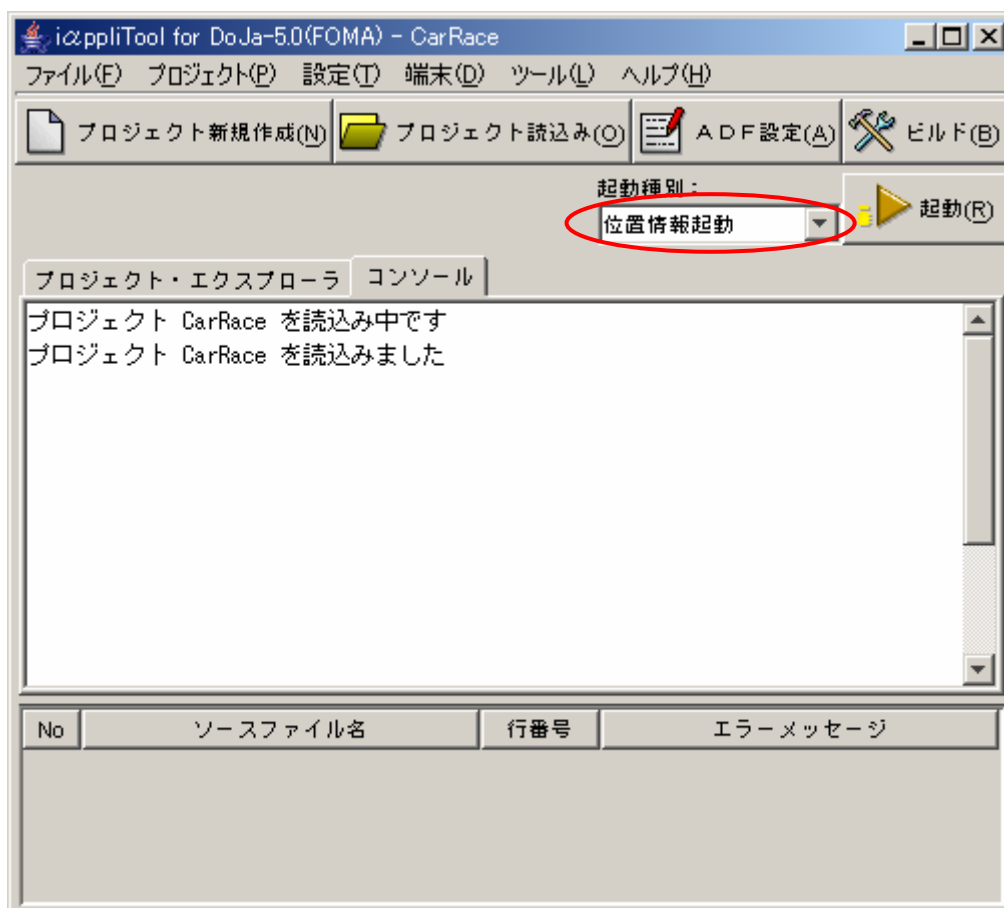


図 4-46 位置情報起動準備

2. 「起動」ボタンを押します。

位置情報エントリID入力画面が表示されます（図 4-4 7）。このとき、エミュレータが以下の状態の時には位置情報エントリID入力画面は表示されません：

- メーラ（「4.1 6. メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。
- コードリーダー（「4.1 8. コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

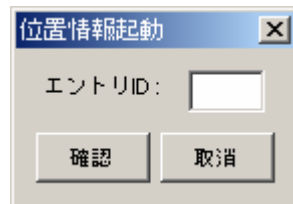


図 4-4 7 位置情報エントリID入力画面

3. 位置情報編集で登録した位置情報のエントリIDを入力して「確認」を押します。
4. 位置情報起動確認画面が表示されるので、「起動」を押します（図 4-4 8）。

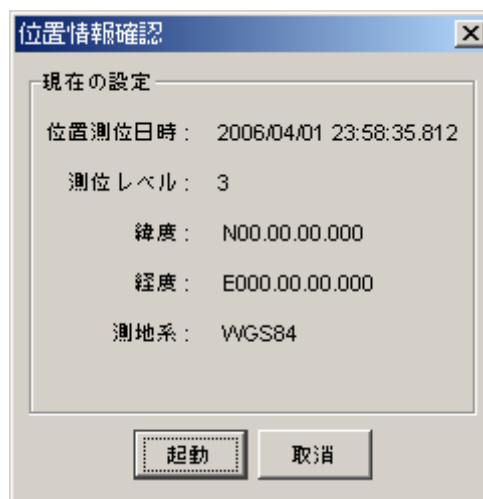


図 4-4 8 位置情報起動確認画面

5. トラステッドであるiアプリを起動してもよいか確認画面が表示されるので、「はい」を押します（図 4-4 9）。

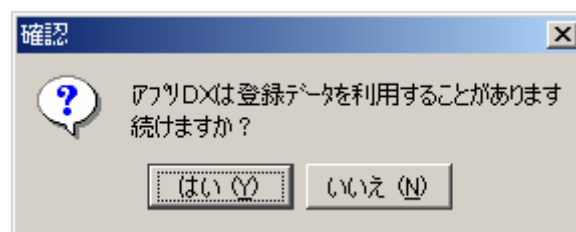


図 4-4 9 トラステッド起動可否

6. 位置情報のパラメータが渡ってアプリケーションが起動します（図 4-5 0）。



図 4-5 0 位置情報起動されたアプリケーション

渡される位置情報のパラメータは以下のとおり：

パラメータ (IApplication#getParameter)	内容
"latitude"	緯度
"longitude"	経度
"datum"	測地系 (WGS84:0 Tokyo datum:1)
"level"	測位レベル (1, 2, 3)
"timestamp"	測位日時 ()
"entry-id"	画像、電話帳のエントリ ID

4.20. 位置情報画像起動

「位置情報画像起動」とは、i アプリケーションを位置情報が登録された画像（地図 等）から起動する方法で「通常起動」とは以下の点が異なります：

- トラステッド i アプリです（「5.2. トラステッド動作の設定」参照）。
- 「位置情報画像起動」の起動タイプは「LAUNCHED_FROM_LOCATION_IMAGE」です。
- ADFの設定項目「TrustedAPID」に適切な値を設定し、「トラステッド動作設定」→「位置情報取得APIの許可」を yes にした場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

4.20.1. 位置情報画像起動手順

1. メイン画面のプルダウンメニューから「位置情報画像起動」を選択します（図 4-5 1）。

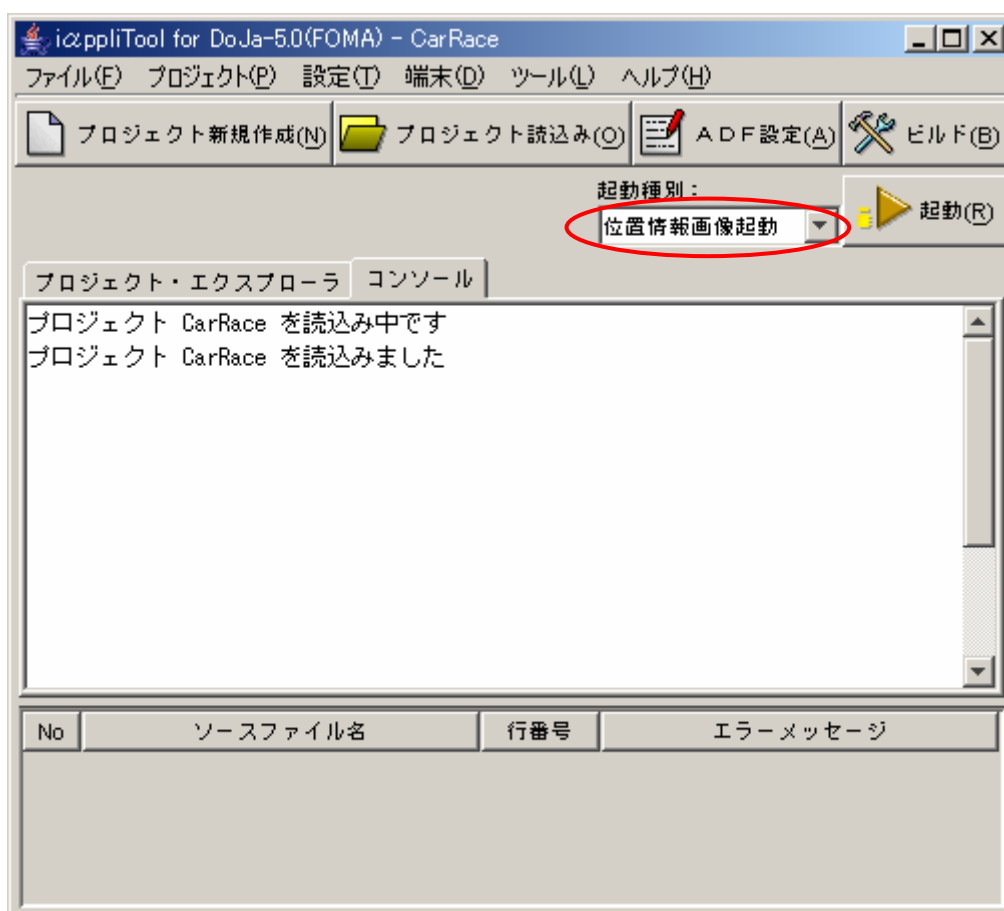


図 4-5 1 位置情報画像起動準備

2. 「起動」ボタンを押します。

位置情報画像エントリID入力画面が表示されます（図 4-5 2）。このとき、エミュレータが以下の状態の時には位置情報画像エントリID入力画面は表示されません：

- メーラ（「4.1 6. メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。
- コードリーダー（「4.1 8. コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

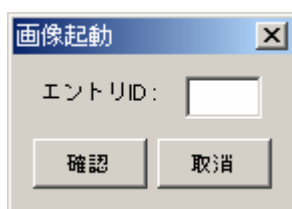


図 4-5 2 位置情報画像エントリID入力画面

3. 位置情報編集で登録した画像のエントリIDを入力して「確認」を押します。
4. 位置情報起動確認画面が表示されるので、「起動」を押します（図 4-5 3）。

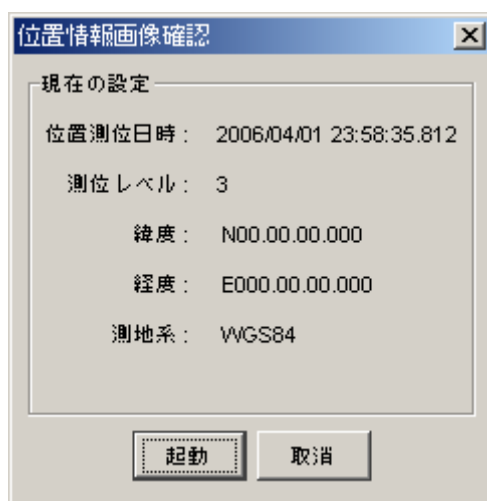


図 4-5 3 位置情報画像起動確認画面

5. トラストッドであるiアプリを起動してもよいか確認画面が表示されるので、「はい」を押します（図 4-5 4）。

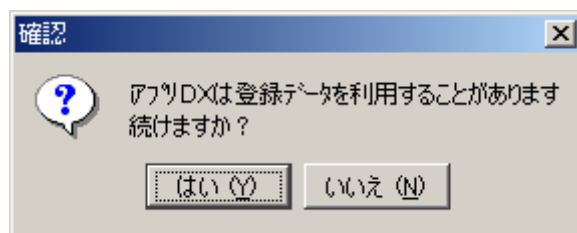


図 4-5 4 トラストッド起動可否

6. 位置情報のパラメータが渡ってアプリケーションが起動します（図 4-5 5）。



図 4-5 5 位置情報画像起動されたアプリケーション

4.21. 位置情報電話帳起動

「位置情報電話帳起動」とは、i アプリを位置情報が登録された電話帳から起動する方法で「通常起動」とは以下の点が異なります：

- トラステッドi アプリです（「5.2. トラステッド動作の設定」参照）。
- 「位置情報電話帳起動」の起動タイプは「LAUNCHED_FROM_PHONEBOOK」です。
- ADFの設定項目「TrustedAPID」に適切な値を設定し、「トラステッド動作設定」→「位置情報取得APIの許可」をyesにした場合のみ起動可能となります（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

4.21.1. 位置情報電話帳起動手順

1. メイン画面のプルダウンメニューから「位置情報電話帳起動」を選択します（図4-46）。

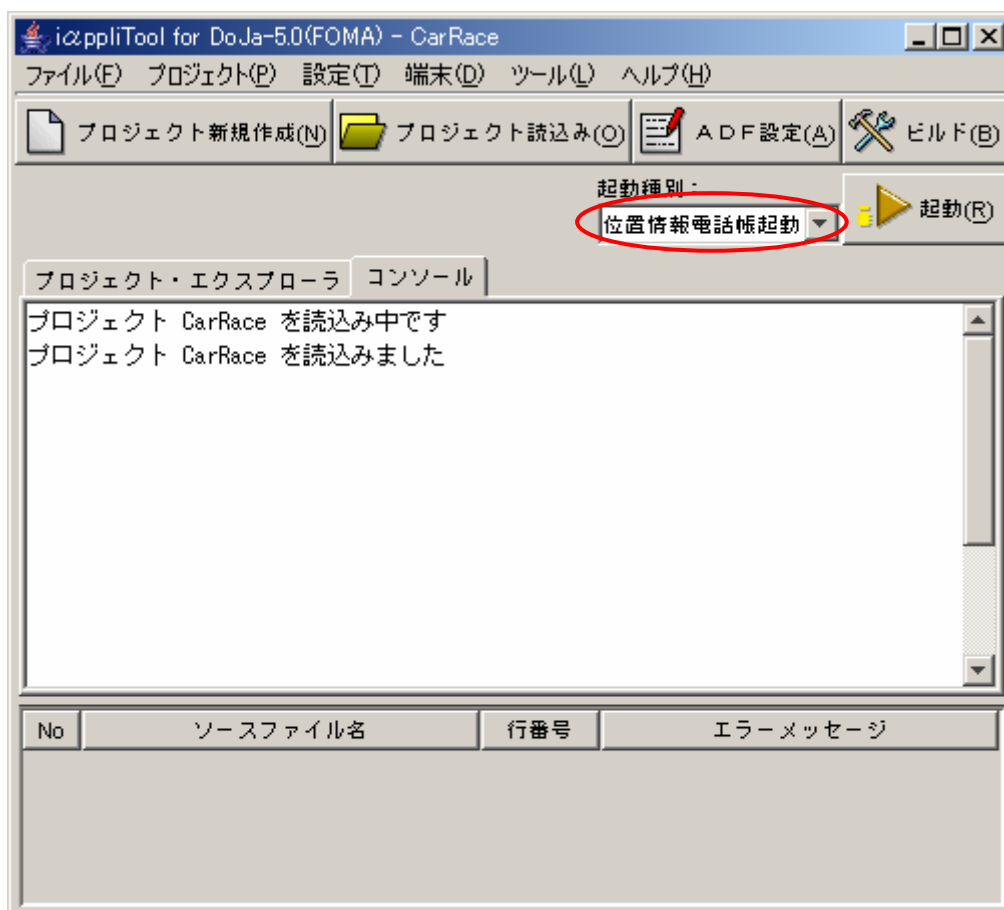


図 4-56 位置情報電話帳起動準備

2. 「起動」ボタンを押します。

電話帳エントリID入力画面が表示されます（図 4-5 7）。このとき、エミュレータが以下の状態の時には電話帳エントリID入力画面は表示されません：

- メーラ（「4.1 6. メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。
- コードリーダー（「4.1 8. コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

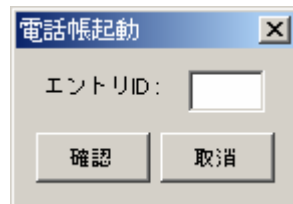


図 4-5 7 電話帳エントリID入力画面

3. 位置情報編集もしくは電話帳の編集で、位置情報を登録した電話帳のエントリIDを入力して「確認」を押します。
4. 位置情報電話帳起動確認画面が表示されるので、「起動」を押します（図 4-5 8）。

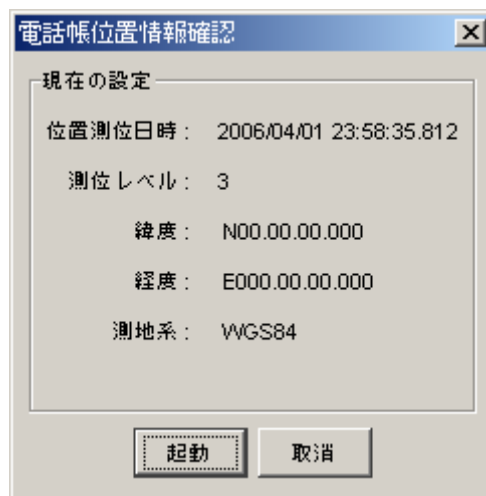


図 4-5 8 位置情報電話帳起動確認画面

5. トラステッドである i アプリを起動してもよいか確認画面が表示されるので、「はい」を押します (図 4-5 9)。

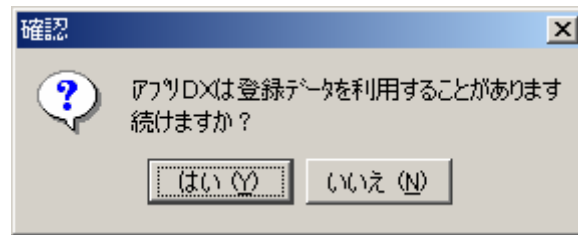


図 4-5 9 トラステッド起動可否

6. 電話帳に登録された位置情報のパラメータが渡ってアプリケーションが起動します (図 4-6 0)。



図 4-6 0 位置情報電話帳起動されたアプリケーション

4.22. アドホック起動

「アドホック起動」とは、i アプリを FeliCa のアドホック通信から起動する方法で「通常起動」とは以下の点が異なります：

- 「アドホック起動」の起動タイプは「LAUNCHED_FROM_FELICA_ADHOC」です。
- ADF の「AllowPushBy」に「icc:コマンド名」が指定されていなくてはなりません。

エミュレータでは、TCP/IP 上の別のエミュレータ同士で FeliCa アドホック通信を仮想的に実現する事ができます。FeliCa アドホックを実行する上での通信の設定については、「5.10. 仮想 FeliCa 設定」を参照してください。

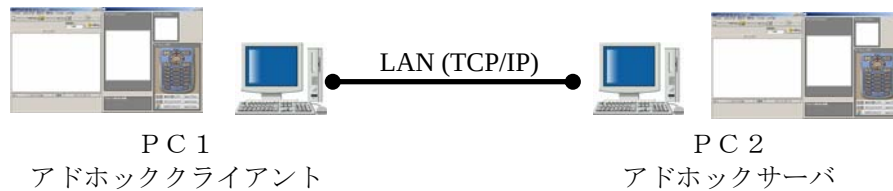


図 4-6 1 仮想 FeliCa アプリのアドホック通信シミュレーション

4.22.1. アドホック起動手順

1. メイン画面のプルダウンメニューから「アドホック起動」を選択します（図 4-6 2）。

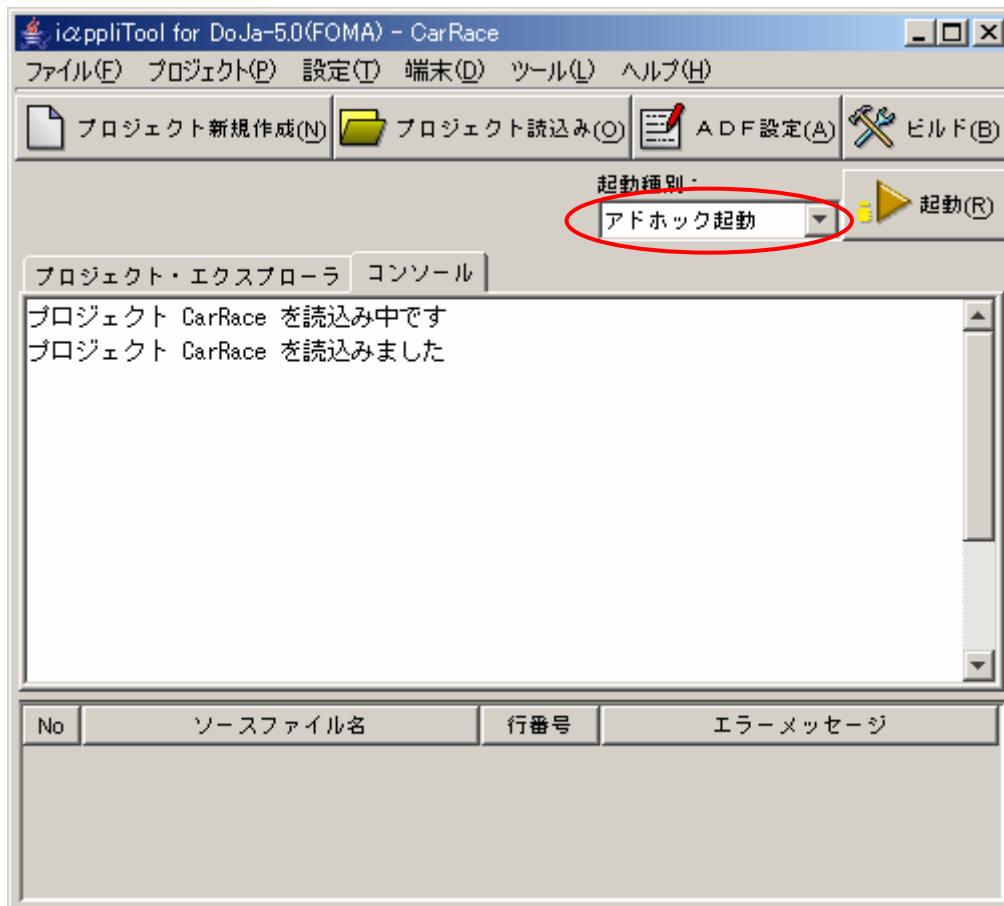


図 4-6 2 アドホック起動準備

2. 「起動」ボタンを押します。
3. アドホック連携待機中になり、仮想FeliCa設定で設定した接続先クライアントからの接続(AdhocDataTransfer#setup())を待ち受けます（図 4-6 3）。



図 4-6 3 アドホック通信待機中画面

4. クライアントからの接続が確立すると「通信中」となり、FelicaAdhocListener#requestReceived() が呼ばれますので、スレッドを起動してFelicaServerObexConnection にてアドホック通信を開始してください。

4.22.2. iアプリが起動中の場合

既に起動している i アプリに対し、アドホック接続を待ち受けるには、エミュレーション端末の右クリックメニューから「FeliCaアドホック接続待ち」を選択します（図 4-6 4）。

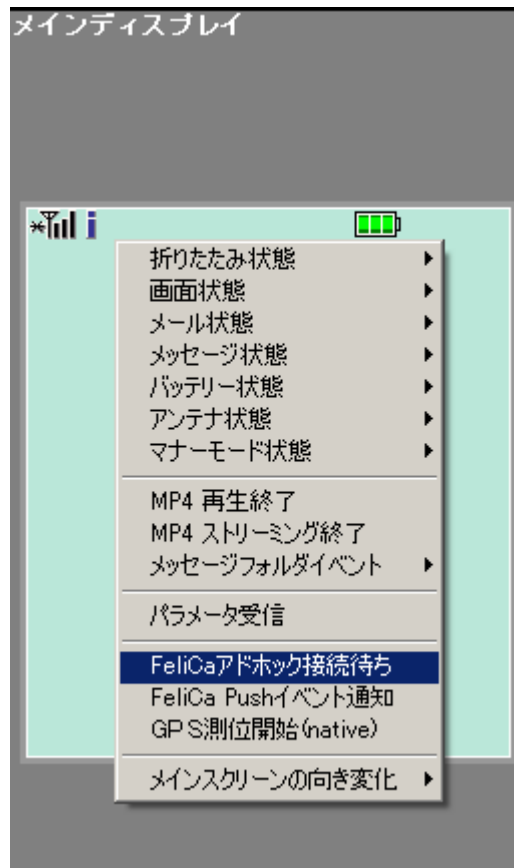


図 4-6 4 FeliCa アドホック接続待ち

コンソール上に「アドホック連携待機開始しました」と表示され、「アドホック連携待機終了しました」と表示されるまでの 1 分間、相手からの接続(AdhocDataTransfer#setup())を待ち受けます。

4.23. トルカ起動

「トルカ起動」とは、i アプリをトルカから起動することです。

「トルカ起動」の起動タイプは、「LAUNCHED_FROM_TORUCA」です。

4.23.1. トルカ起動手順

1. メイン画面のプルダウンメニューから「トルカ起動」を選択します（図 4-6 5）。

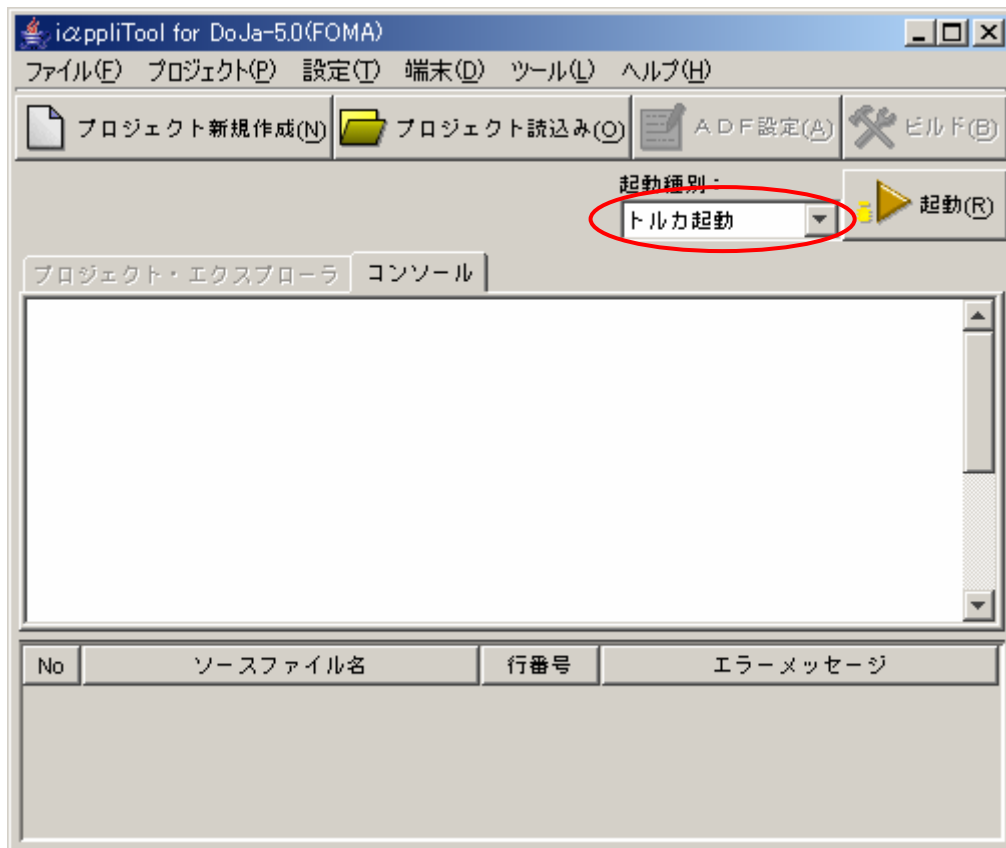


図 4-6 5 トルカ起動準備

2. 「起動」ボタンを押します。

トルカ選択画面が表示されます（図 4-6 6）。このとき、エミュレータが以下の状態の時にはトルカ選択画面は表示されません：

- 既にトルカが起動している。
- メーラ（「4.1 6 メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。
- コードリーダー（「4.1 8 コードリーダー起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML起動」を参照）が起動している。
- DTV（「4.2 6. DTV起動」を参照）が起動している。

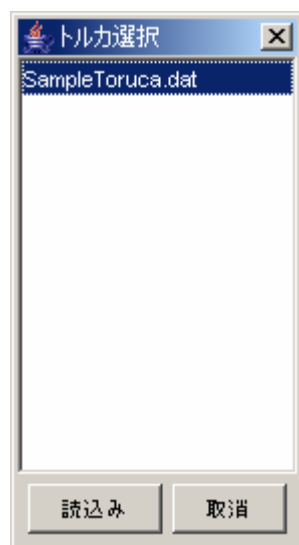


図 4-6 6 トルカ選択画面

3. 選択したトルカのトルカ画面が表示されます（図 4-6 7）。
（この画面は実際のトルカとは異なります。）

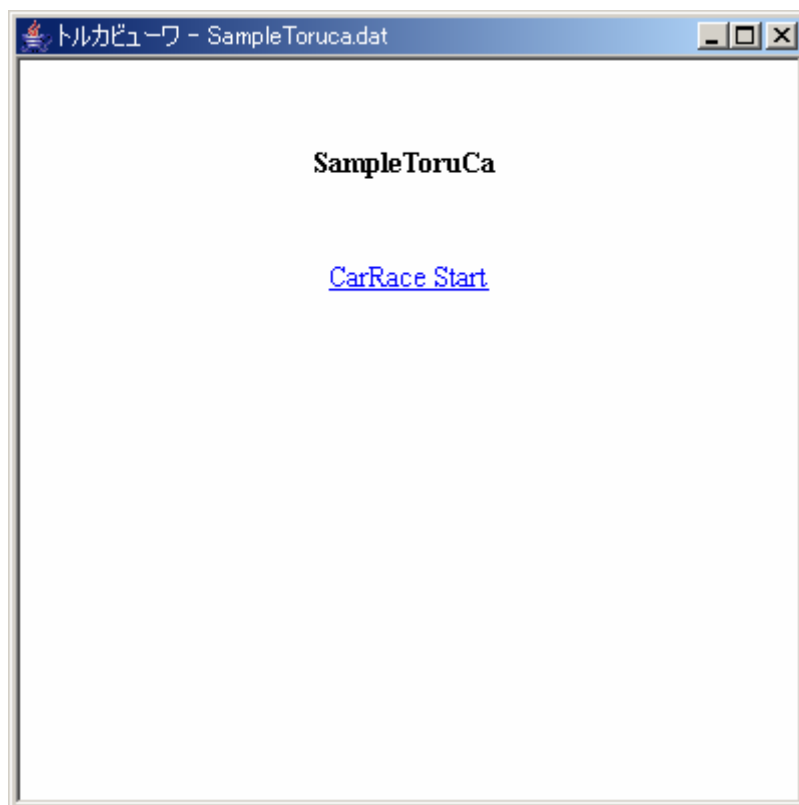


図 4-6 7 トルカ画面

4. 図 4-6 7 のリンク（青い文字）をクリックします。

i アプリをトルカから起動するためには、以下の条件を満たしている必要があります：

- ADFの項目で「LaunchByToruCa」に「any」あるいは起動を許可するURLが設定されていること（「4.7.アプリケーションディスクリプタファイル（ADF）の編集」を参照）。
- トルカファイルのBody部(本文)に起動するアプリが正しく指定されていること（「10.15.トルカファイル」参照）。トルカスニップでは起動できません。
- i アプリが「プロジェクト読み込み画面」で指定したプロジェクト格納先に作成されていること。必ずしもデフォルトの格納先（「C:\iDKDoJa5.0\apps」）に作成されている必要はありません。

上記の条件を満たしていない場合には、リンクが押せないか、i アプリの起動を禁止するダイアログが表示されます。

上記の条件を満たしている場合には、i アプリをトルカ起動するための確認ダイアログが表示され（図 4-68）、「はい」ボタンを押すとi アプリが起動します（図 4-69）。

トルカから待ち受けアプリケーションを起動した場合、待ち受けアプリケーションとしては動作せずに通常のアプリケーションとして動作します。

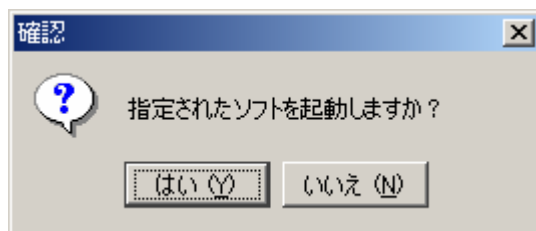


図 4-68 起動確認ダイアログ

トルカデータの作成については、（「5.21.トルカデータ編集」および「10.15.トルカファイル」）を参照してください。



図 4-6 9 トルカ起動された i アプリ

5. トルカ起動された i アプリの終了方法により、その後、自動起動する i アプリが異なります:

- トルカ起動された i アプリを正常終了 (i アプリが `IApplication#terminate()` メソッドを呼び出して終了した場合)

再びトルカ画面が表示されます。トルカ画面を終了する (画面の右上の「×」を押す) と、トルカを起動する際に待ち受けアプリケーションが動作していたのであればそれを起動し、そうでなければ通常の待ち受け画面に戻ります。

- トルカ起動された i アプリを強制終了 (端末の「終話ボタン」を押した場合)
トルカ画面は表示されず、トルカを起動する際に待ち受けアプリケーションが動作していたのであればそれを起動し、そうでなければ通常の待ち受け画面に戻ります。

4.24. 削除時起動

「削除時起動」とは、FeliCa を利用する i アプリを削除するときに起動する場合で「通常起動」とは以下の点が異なります：

- 「削除時起動」の起動タイプは「LAUNCHED_FROM_MENU_FOR_DELETION」です。
- FeliCa 機能を使う i アプリのみ有効な起動種別です。
i アプリは、この起動タイプで起動されたら FeliCa IC チップ内の後処理(退会手続きなど)をしてください。

起動の際の制約などに関しては「通常起動」の場合と同様ですが、実際の携帯端末において FeliCa を利用する i アプリ以外は、この起動種別で起動されることはありません（「4.3. プロジェクトのビルドと通常起動」参照）。

4.24.1. 削除時起動手順

1. メイン画面のプルダウンメニューから「削除時起動」を選択します（図 4-70）。

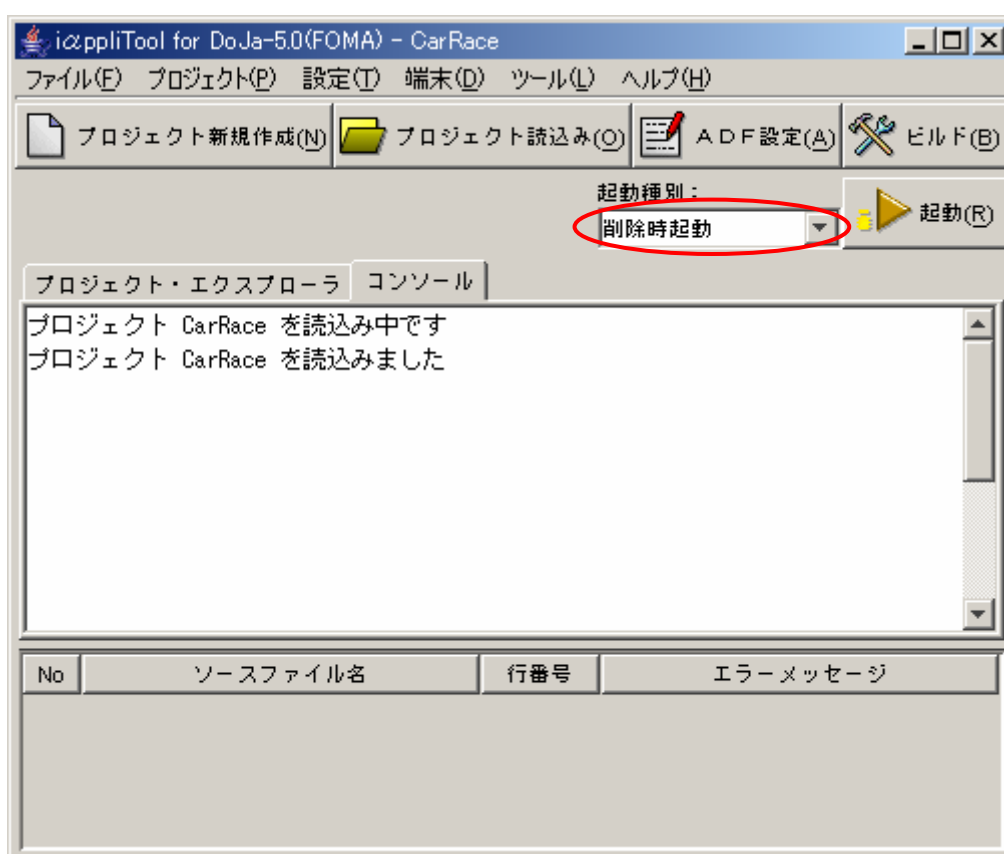


図 4-70 削除時起動準備

2. 「起動」ボタンを押します。
※ エミュレータでは実際に i アプリは削除されません。

4.25. BML起動

「BML 起動」とは、i アプリをBMLブラウザから起動することです。
「BML 起動」の起動タイプは、「LAUNCHED_FROM_BML」です。
この機能は、D T Vに対応した携帯端末からのみ利用可能です。

4.25.1. BML起動手順

エミュレータで i アプリをBMLブラウザから起動するには：

1. メイン画面のプルダウンメニューから「BML 起動」を選択します。（図 4-7 1）。

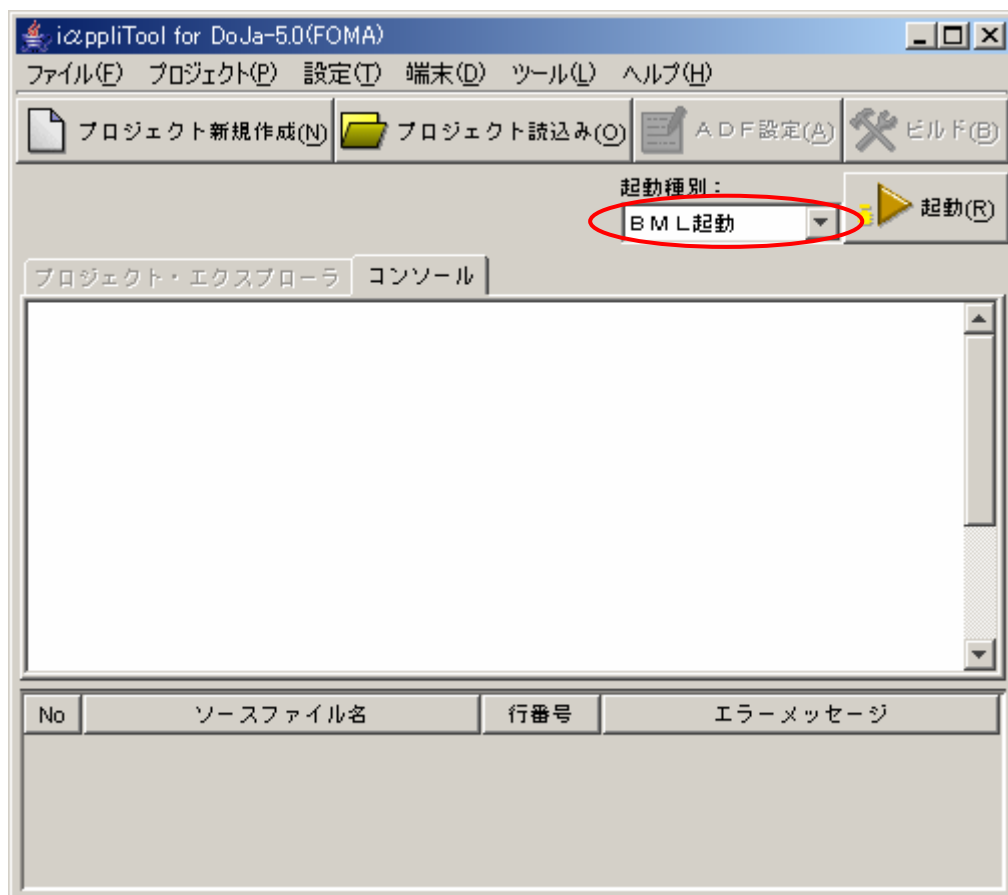


図 4-7 1 BML起動準備

2. 「起動」ボタンを押します。

BMLブラウザ画面が表示されます（図 4-7 2）。このとき、エミュレータが以下の状態の時にはBMLブラウザ画面は表示されません：

- 既にBMLブラウザが起動している。
- メーラ（「4.1 6 メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。

- コードリーダー（「4.1 8 コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3.トルカ起動」を参照）が起動している。
- DTV（「4.2 6.DTV起動」を参照）が起動している。

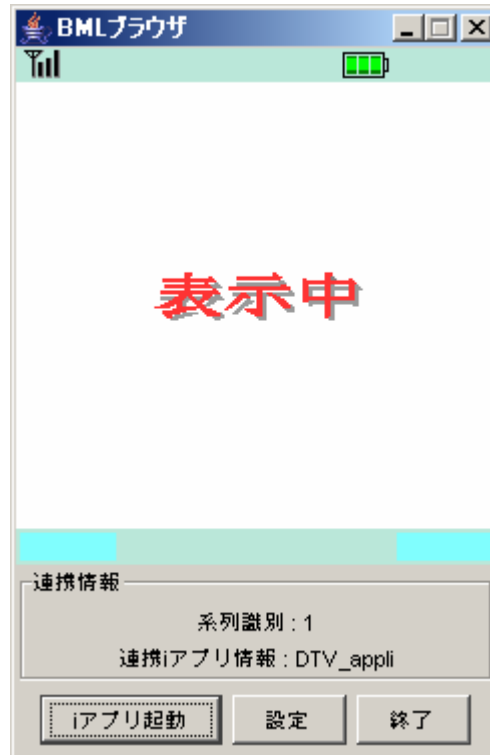


図 4-7 2 BMLブラウザ画面

3. 連携情報を設定したい場合は、BMLブラウザ画面で「設定」ボタンを押します。

連携情報設定画面が表示されます（図 4-7 3）。

この画面では、BMLブラウザの系列識別・連携先 i アプリ jam ファイル名・連携パラメータを設定することが可能です。

系列識別：

表示中のBMLコンテンツの系列識別として、1～11を選択可能です。

連携先 i アプリ jam ファイル名：

連携先 i アプリとして、現在のプロジェクト格納先に存在する i アプリのうち、「ADF 設定」→「LaunchByBML」が設定されている i アプリを選択可能です。

連携パラメータ：

連携先 i アプリにおいて IApplication#getParameter("bml_param")で取得するパラメータを設定可能です。省略した場合は上記メソッドにてNULLが取得されます。



図 4-7 3 連携情報設定画面

設定内容を反映する時は「設定」ボタンを、破棄する場合は「取消」ボタンを押します。押下後、BMLブラウザ画面（図 4-7 2）に戻ります。

4. BMLブラウザ画面（図 4-7 2）で「i アプリ起動」ボタンを押します。
i アプリをBMLブラウザから起動するためには、以下の条件を満たしている必要があります：

- 現在のプロジェクト格納先に連携先 i アプリが格納されていること。
- 連携先 i アプリの「ADF設定」→「LaunchByBML」（「4.7」を参照）に設定されている系列識別とBMLブラウザ画面に設定されている系列識別が一致すること。

上記の条件を満たしていない場合には、i アプリの起動を禁止するダイアログあるいはコンソールにエラーメッセージが表示されます。

上記の条件を満たしている場合には、i アプリをBMLブラウザから起動するための確認ダイアログが表示され「はい」ボタンを押すと i アプリが起動します（図 4-7 4）。



図 4-7 4 BML起動された i アプリ

4.26. DTV起動

「DTV起動」とは、番組表 i アプリをDTVから起動することです。
「DTV起動」の起動タイプは、「LAUNCHED_FROM_DTV」です。
この機能は、DTVに対応した携帯端末からのみ利用可能です。

4.26.1. DTV起動手順

エミュレータで i アプリをDTVから起動するには：

1. メイン画面のプルダウンメニューから「DTV起動」を選択します（図 4-7 5）。

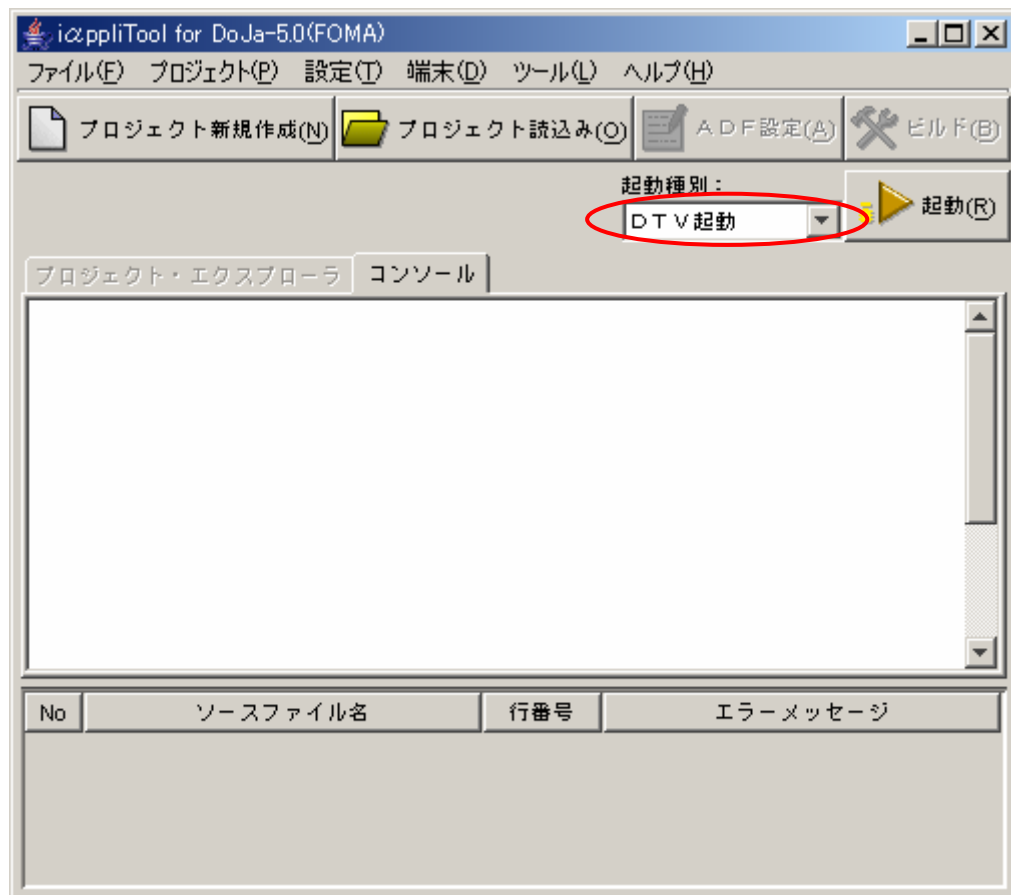


図 4-7 5 DTV起動準備

2. 「起動」ボタンを押します。

DTV画面が表示されます（図 4-7 6）。このとき、エミュレータが以下の状態の時にはDTV画面は表示されません：

- 既にDTVが起動している。
- メーラ（「4.1 6 メーラ起動」を参照）が起動している。
- ブラウザ（「4.1 7. ブラウザ起動」を参照）が起動している。

- コードリーダー（「4.1 8 コードリーダー起動」を参照）が起動している。
- トルカ（「4.2 3.トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.2 5. BML 起動」を参照）が起動している。

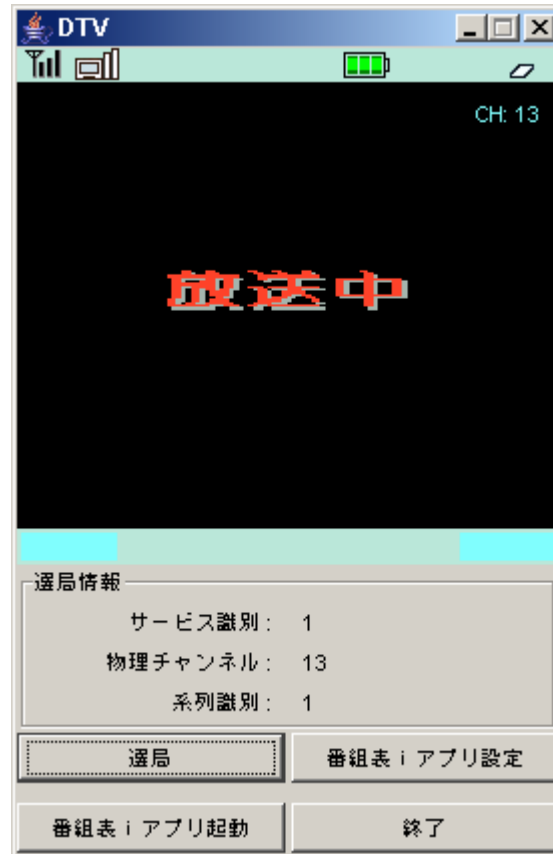


図 4-7 6 D T V画面

3. 選局情報を設定したい場合は、D T V画面で「選局」ボタンを押します。

選局画面が表示されます（図 4-7 7）。この画面では、D T Vのサービス識別・物理チャンネル・系列識別 を設定することが可能です。

サービス識別：

表示中のD T Vのサービス識別を設定可能です。

連携先 i アプリにおいて IApplication#getParameter("service_id")で取得するパラメータとなります。省略した場合は上記メソッドにてN U L Lが取得されます。

物理チャンネル：

表示中のD T Vの物理チャンネルを設定可能です。

ここで設定した値は i アプリの動作には影響しません。

系列識別：

表示中のD T Vの系列識別を設定可能です。

ここで設定した値は i アプリの動作には影響しません。

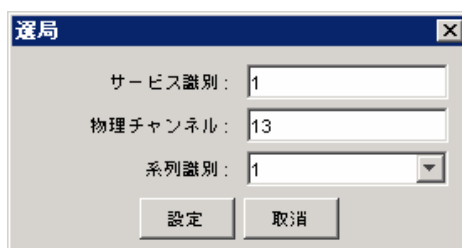


図 4-7 7 選局画面

設定内容を反映する時は「設定」ボタンを、破棄する場合は「取消」ボタンを押します。押下後、DTV画面（図 4-7 6）に戻ります。

4. 番組表 i アプリ（連携先 i アプリ）を設定したい場合は、DTV画面で「番組表 i アプリ設定」ボタンを押下します。

番組表 i アプリ設定画面が表示されます（図 4-7 8）。この画面では、番組表 i アプリ ADF ファイル名を設定することが可能です。

番組表 i アプリ ADF ファイル名：

連携先 i アプリとして、現在のプロジェクト格納先に存在する i アプリのうち、「ADF 設定」→「TrustedAPID」が設定されており、かつ「トラステッド動作設定」→「番組表 i アプリ起動の許可」が yes に設定されている i アプリを選択可能です。



図 4-7 8 番組表 i アプリ設定画面

設定内容を反映する時は「設定」ボタンを、破棄する場合は「取消」ボタンを押します。押下後、DTV画面（図 4-7 6）に戻ります。

5. DTV画面（図 4-7 6）で「番組表 i アプリ起動」ボタンを押します。
i アプリをDTVから起動するためには、以下の条件を満たしている必要があります：

- 現在のプロジェクト格納先に連携先 i アプリが格納されていること。
- 「ADF設定」→「TrustedAPID」が設定されていること（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」を参照）。
- 「トラステッド動作設定」→「番組表 i アプリ起動の許可」が yes に設定されていること（「5.2. トラステッド動作の設定」を参照）。

上記の条件を満たしていない場合には、i アプリの起動を禁止するダイアログあるいはコンソールにエラーメッセージが表示されます。

上記の条件を満たしている場合には、i アプリをD T Vから起動するための確認ダイアログが表示され「はい」ボタンを押すと i アプリが起動します（図 4-7 9）。



図 4-7 9 DTV起動された i アプリ

4.27. 各種起動方法による画面遷移

これまで「iα ppliTool」での i アプリの起動方法には、以下のような方法があると説明してきました：

- 通常起動
- 待ち受け起動
- 初回起動
- タイマ起動
- 外部機器起動
- 即起動
- 受信フォルダ起動
- 送信フォルダ起動
- 未送信フォルダ起動
- メーラ起動
- ブラウザ起動
- コードリーダー起動
- 位置情報起動
- 位置情報画像起動
- 位置情報電話帳起動
- アドホック起動
- トルカ起動
- 削除時起動
- BML 起動
- D T V 起動

ここで、これらを組み合わせた場合の画面遷移を図 4-8 0 に示します。

「iα ppliTool」実行中、画面に表示されている i アプリは、常に 1 つだけです。

また、それぞれ以下に示す条件で起動する事が可能です：

- メーラが起動していない
- ブラウザが起動していない
- コードリーダーが起動していない
- トルカが起動していない
- BML ブラウザが起動していない
- D T V が起動していない

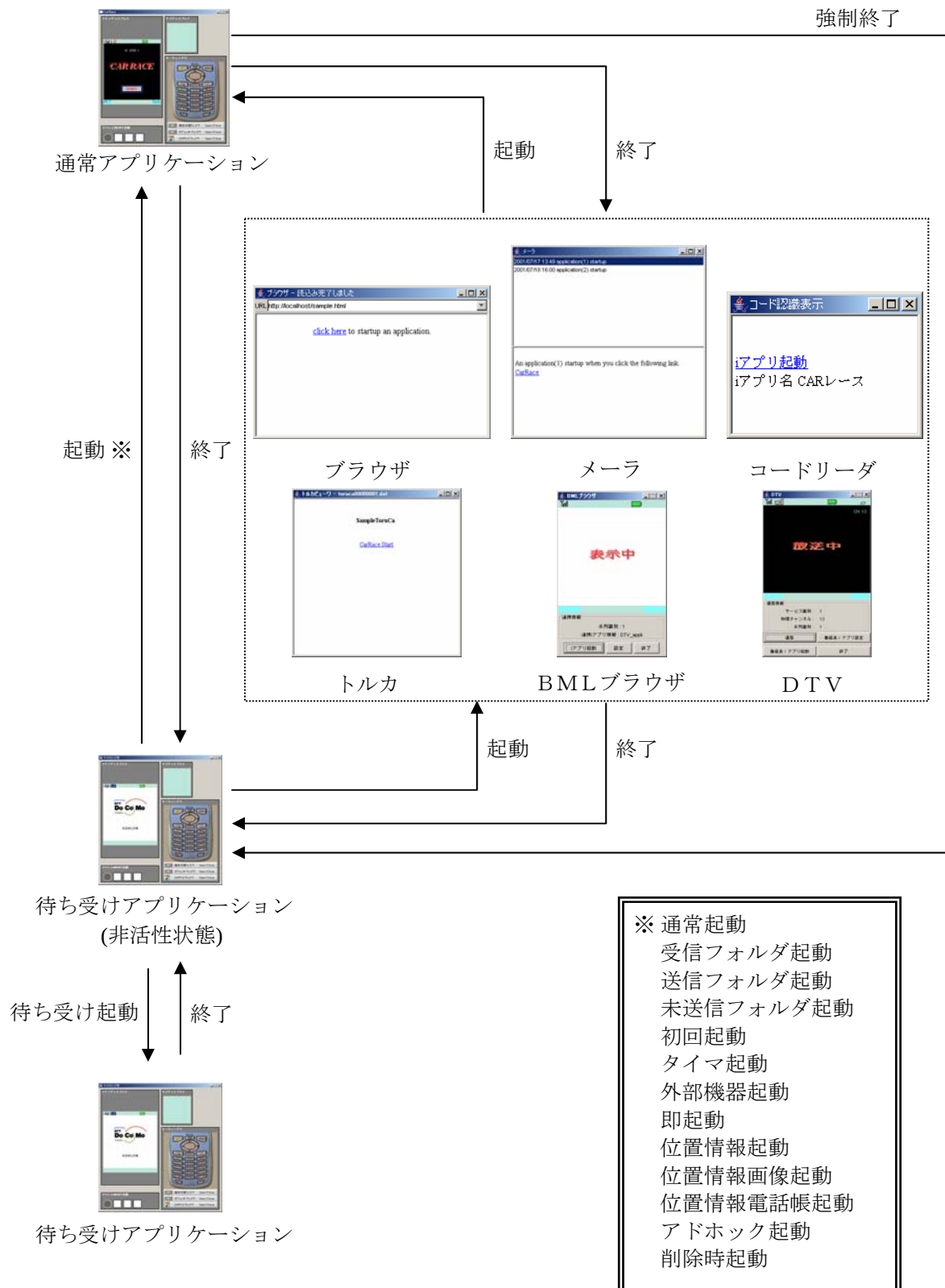


図 4-80 「iα ppliTool」における画面遷移

第5章. エミュレータの設定

この章では、エミュレータのいろいろな設定について説明します。

5.1. ネットワークの設定

HTTP通信および赤外線通信のネットワークの設定をします。

ネットワークの設定をするには、

1. メニューから「設定」→「アプリケーション動作環境設定...」→「ネットワーク設定」タブを選択します（図 5-1）。

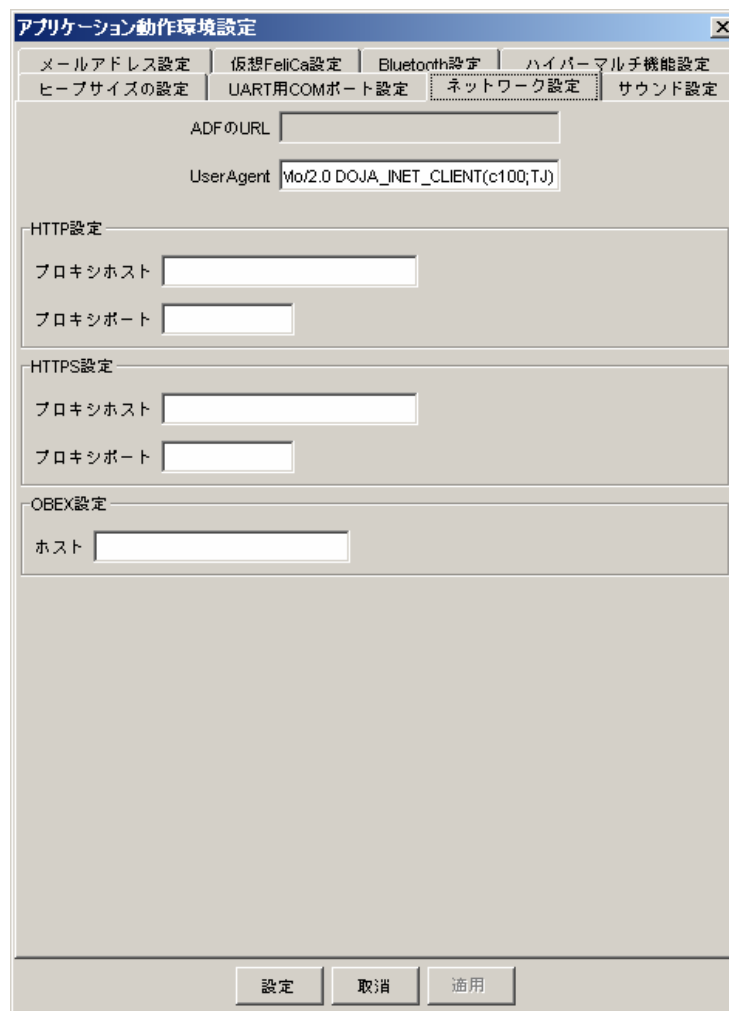


図 5-1 ネットワークの設定画面

2. 各設定項目を設定し、「設定」もしくは「適用」を押します。

各設定項目は表 5-1 に示すとおりです。

表 5-1 エミュレータの設定項目

設定項目		説明	デフォルト
ADF の URL		i アプリの ADF が存在する URL。 PackageURL タグに相対 URL が指定された場合、本 URL がベースの URL となります。i アプリはこの PackageURL のホストとポートのみ通信可能です。 このフィールドに指定する URL は、ADF の PackageURL タグのフォーマットと同様です。 また、このフィールドは、アプリケーションごとに設定可能なので、「iα ppliTool」でプロジェクトが読み込まれている場合に設定可能です。さらに、このフィールドに設定した URL は、ファイルに保存されるので、次回「iα ppliTool」起動時に自動的に設定されます。	なし
UserAgent		HTTP の「UserAgent」ヘッダ。 このフィールドに設定した「UserAgent」ヘッダは、レジストリに保存されるので、次回「iα ppliTool」起動時に自動的に設定されます。	DoCoMo/2.0 DOJA_INET_CLIENT(c100;TJ)
HTTP 設定	プロキシ ホスト	i アプリ及び「iα ppliTool」のブラウザが Web サーバとの通信に使用する HTTP プロキシサーバのホスト名。「5.1.1. プロキシサーバの設定」を参照。 また、このフィールドに設定したホスト名は、レジストリに保存されるので、次回「iα ppliTool」起動時に自動的に設定されます。	なし
	プロキシ ポート	i アプリおよび「iα ppliTool」のブラウザが Web サーバとの通信に使用する HTTP プロキシサーバのポート番号。「5.1.1. プロキシサーバの設定」を参照。 また、このフィールドに設定したポート番号は、レジストリに保存されるので、次回「iα ppliTool」起動時に自動的に設定されます。	なし

HTTPS 設定	プロキシ ホスト	i アプリがWebサーバとの通信に使用するHTTPSプロキシサーバのホスト名。「5.1.1.1. プロキシサーバの設定」を参照。 また、このフィールドに設定したホスト名は、レジストリに保存されるので、次回「iアプリTool」起動時に自動的に設定されます。	なし
	プロキシ ポート	i アプリがWebサーバとの通信に使用するHTTPSプロキシサーバのポート番号。「5.1.1.1. プロキシサーバの設定」を参照。 また、このフィールドに設定したポート番号は、レジストリに保存されるので、次回「iアプリTool」起動時に自動的に設定されます。	なし
OBEX 設定	ホスト	i アプリがOBEXを使用して通信する際のIPアドレス。OBEX over TCPを用いる場合に、接続先ノードのIPアドレスを指定します。指定がない場合は、127.0.0.1（ローカルホスト）にコネクションを確立する試みをします。なお、本設定は、ObexClientConnectionクラスのconnectメソッドで意味をなします。「第12章. OBEXの利用」を参照。 また、このフィールドに設定したホスト名は、レジストリに保存されるので、次回「iアプリTool」起動時に自動的に設定されます。	なし

5.1.1. プロキシサーバの設定

エミュレータのもとで動作する i アプリからプロキシサーバ経由で Web サーバに接続する必要がある場合には、エミュレータの設定でプロキシサーバの情報を設定する必要があります。i アプリは、HTTP か HTTPS を使用して Web サーバに接続しますが、同じ i アプリで両方のプロトコルを使用することはできません。

HTTPとHTTPSのホストとポートは、図 5-1 ネットワークの設定画面で設定します。

なお、HTTPSの際に使用する証明書の管理については「第11章. 証明書の管理」を参照してください。

5.2. トラステッド動作の設定

トラステッド動作は、i アプリがトラステッドであるAPIを利用する**トラステッドiアプリ（iアプリDX）**である場合に設定します。

トラステッドiアプリでは、セキュリティ上、通常のiアプリで禁じられている以下のような機能を実現することができます。

- 携帯電話本体に登録されたデータ（電話帳、発着信履歴等）との連携
- iアプリダウンロード元以外のサーバとの通信
- iモードメールとの連携
- 位置情報を使った機能
- DTV アプリからの番組表iアプリ起動機能
- OpenGL/ESを使った機能
- FeliCaを使った機能（一部を除く）

iモードメニューに掲載されていないコンテンツ提供者は、トラステッドiアプリ（iアプリDX）機能を利用したiアプリコンテンツを配信することは出来ません。

トラステッド動作を設定するには：

1. メニューから「設定」→「トラステッド動作設定...」を選択します。
「トラステッド動作設定画面」が表示されます（図 5-2）。

トラステッド動作設定	
電話帳/発信着信履歴/未読メール参照機能の許可	☐ Yes ☒ No
ネイティブメール利用の許可	☐ Yes ☒ No
メッセージアプリ機能の使用許可	☐ Yes ☒ No
テーマ設定の許可	☐ Yes ☒ No
iアプリ自動起動APIの許可	☐ Yes ☒ No
位置情報取得APIの許可	☐ Yes ☒ No
Bluetooth機能の使用許可	☐ Yes ☒ No
番組表iアプリ起動の許可	☐ Yes ☒ No
OpenGL/ES機能の使用許可	☐ Yes ☒ No
通信を許可するURL	
設定 取消	

図 5-2 トラステッド動作設定画面

2. 利用したい機能を Yes に設定します。
「設定」ボタンを押すことにより変更が有効となります。

各設定項目は、表 5-2 に示すとおりです。

表 5-2 トラストッド動作設定の設定項目

設定項目	説明
電話帳/発着信履歴/未読メール参照機能の許可	電話帳の参照、電話発着信履歴の参照、メール受信履歴の参照 をする A P I を利用する場合に yes に変更します。 (関連: PhoneBook, PhoneBookGroup, PhoneBookParam, CallRecord)
ネイティブメーラ利用の許可	ネイティブメーラのメールメニューを利用する場合に yes に変更します。 (関連: Mail, MailAgent, MailDraft)
メッセージ i アプリ機能の使用許可	メッセージ i アプリの A P I を利用する場合に yes に変更します。 (関連: Message, MessageAgent, MessageDigest, MessageReceive, MessageSent)
テーマ設定の許可	着信メロディの保存と設定、発着信画像および待ち受け画面画像の保存と設定 をする A P I を利用する場合に yes に変更します。 (関連: PhoneSystem, SoundStore)
i アプリ自動起動 A P I の許可	アプリケーションの自動起動をする A P I を利用する場合に yes に変更します。 (関連: IApplication#setLaunchTime)
位置情報取得 A P I の許可	位置情報を扱う A P I を利用する場合に yes に変更します。 (関連: Location, LocationProvider, Degree)
Bluetooth 機能の使用許可	Bluetooth の A P I を利用する場合に yes に変更します。 (関連: Bluetooth, BTConnection, SPPConnection, RemoteDevice, BTStateListener)
番組表 i アプリ起動の許可	D T V アプリ (ネイティブ) からの番組表 i アプリ起動機能を利用する場合に yes に変更します。
OpenGL/ES 機能の使用許可	OpenGL/ES の A P I を利用する場合に yes に変更します。(関連: com.nttdocomo.ui.opengl.*)
通信を許可する U R L	ダウンロード元以外に i アプリが通信を許可する U R L を設定します。(最大 255 バイト) ディレクトリの指定は無効で、設定する文字列は、ASCII 形式とし、255 バイトを超えて設定した場合は、i アプリの起動を禁止します。

 ワンポイント

A P I リファレンスにて「トラステッド i アプリ」と指定のある A P I、もしくは、
「**SecurityException** - パーミッションとして〇〇〇の参照が許可されているが、
i アプリ個別のユーザ設定により許可されない場合に発生します。」という例外が
発生する A P I を利用する場合には、この動作を設定する必要があります。

A P I リファレンスは、NTT ドコモの次の URL から入手することができます：
<http://www.nttdocomo.co.jp/service/imode/make/index.html>

5.3. メールアドレスの設定

端末に登録された自分のメールアドレスを設定します。

メールアドレスの設定をするには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「メールアドレス設定」タブを選択します（図 5-3）。

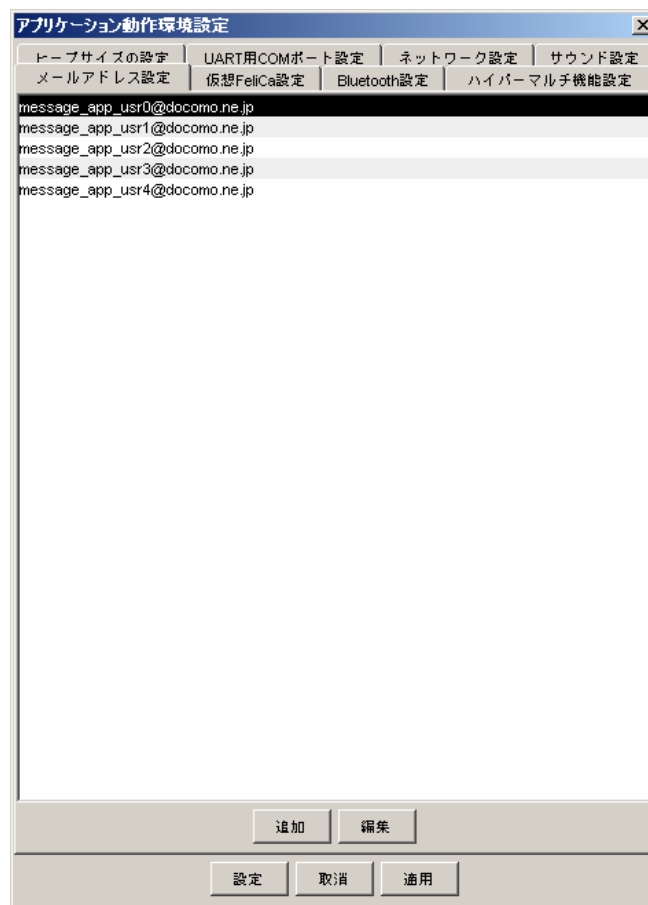


図 5-3 メールアドレスの設定画面

2. メールアドレスを新たに追加する場合は「追加」、変更する場合はメールアドレスを選択して「編集」を押します（図 5-4）。



図 5-4 メールアドレス追加画面

3. メールアドレスを入力して「設定」ボタンを押します。
メールアドレス設定画面の一覧にメールアドレスが追加されます。
4. 移動機に登録するメールアドレスを選択して、「設定」もしくは「適用」ボタンを押します。
デフォルトでは、「message_app_usr0@docomo.ne.jp」が選択されています。
(実際の移動機に登録できるメールアドレスは1つだけです。)

ワンポイント

エミュレータ特有の機能で、ここで作成したメールアドレスに i アプリから送信すると、そのメールアドレスを選択している i アプリでメールを受信する事ができます (受信ボックスに格納されています)。
これにより、エミュレータでも i アプリ同士で相互にメールをやり取りする事ができます。 (「5.16.2.メールの送受信エミュレート機能」を参照)

5.4. リモート制御の設定

プッシュ機能により、サーバから i アプリに渡すパラメータを設定します。
(関連 A P I : PushManager)

パラメータの設定をするには：

1. メニューから「設定」→「リモート制御設定...」を選択します(図 5-5)。

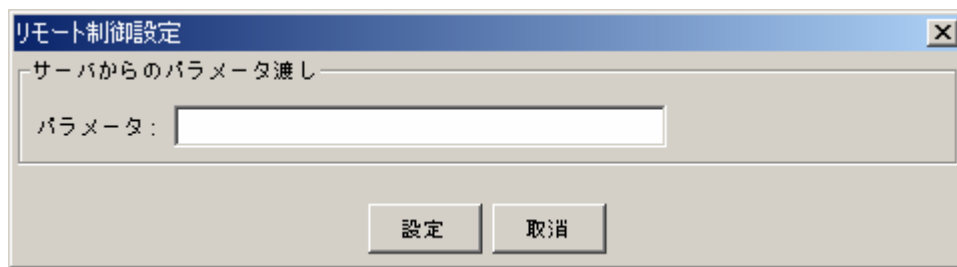


図 5-5 リモート制御パラメータの設定画面

2. パラメータを入力して「設定」ボタンを押します。
「アプリケーション名.properties」ファイルにパラメータが追加されます。

設定したパラメータを i アプリから取得するには：

1. i アプリの起動中に端末ウィンドウを右クリックします。右クリックメニューから「パラメータ受信」を選択すると、パラメータ受信のイベントが発生し、i アプリにてパラメータを取得できます(図 5-6)。

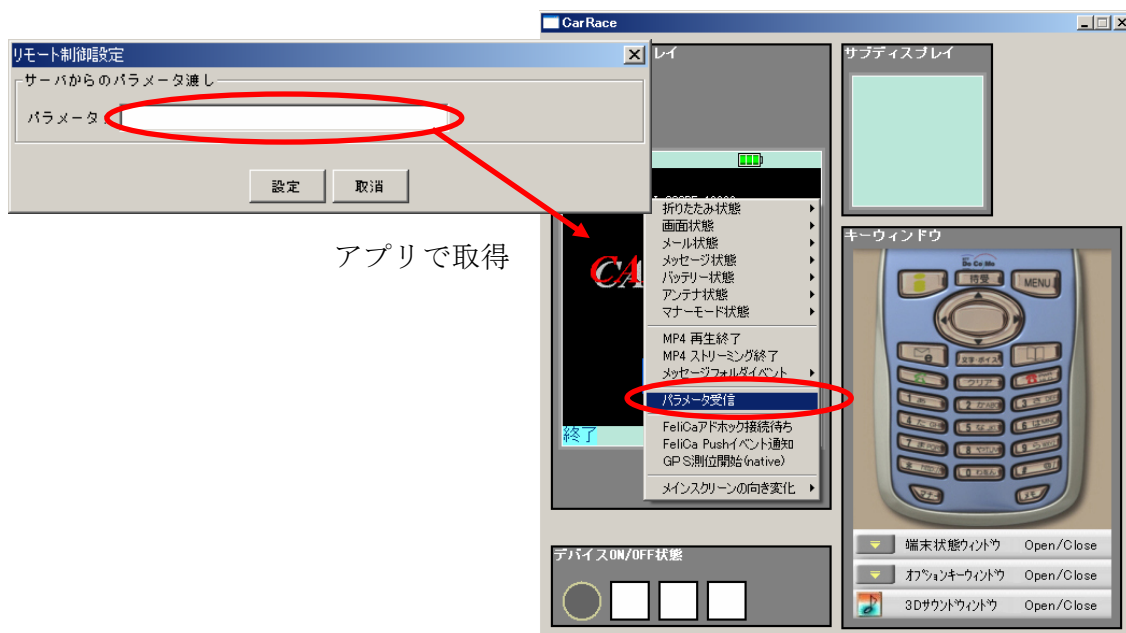


図 5-6 パラメータを i アプリから取得する

5.5. リモート制御のパラメータ領域の削除

「5.4. リモート制御の設定」で設定したパラメータを削除します。

リモート制御のパラメータ領域の削除をするには：

1. メニューから「設定」→「リモート制御のパラメータ領域削除...」を選択すると「アプリケーション名.properties」ファイルからパラメータが削除されます（図5-7）。
これにより、「一度もパラメータを受信していない状態」を試す事ができます。

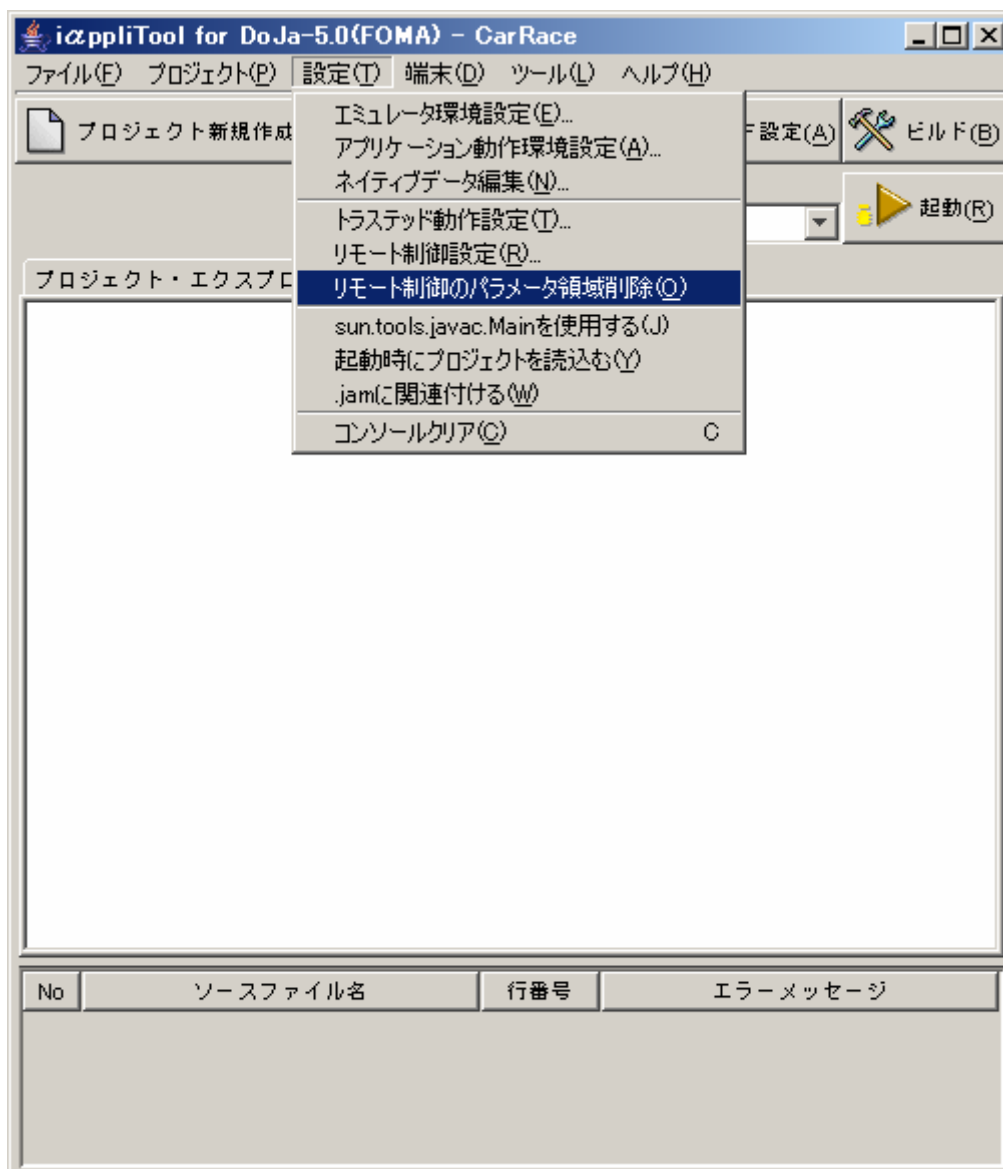


図 5-7 リモート制御パラメータ領域削除

5.6. サウンド設定

サウンドの再生モードの切り替えと、i アプリから利用するMF i のサウンドデータで、UCS 音色など通常のPC上では鳴動できない音の代替音を設定します。

サウンドの再生モードが、「通常再生モード」の場合、**DirectX** を必要とします。もし、**DirectX が入っていない状態で「通常再生モード」に設定した場合は、サウンドを鳴動させるとエラーになります。**

DirectX は、次のサイトから入手できます：

<http://www.microsoft.com/japan/windows/directx/>

サウンド再生の設定をするには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「サウンド設定」タブを選択します（図 5-8）。

The screenshot shows the 'アプリケーション動作環境設定' (Application Environment Settings) dialog box with the 'サウンド設定' (Sound Settings) tab selected. The settings are as follows:

- 再生モード選択** (Playback Mode Selection):
 - 再生モード (Playback Mode): 通常再生モード (Normal Playback Mode)
 - サウンドライブラリ (Sound Library): 汎用サウンドライブラリ (General Sound Library)
- デバイス設定** (Device Settings):
 - デバイス (Device): プライマリ サウンド ドライバ (Primary Sound Driver)
- 代替音設定** (Replacement Sound Settings):
 - チャンネル (Channel): チャンネル 1 (Channel 1)
 - 楽器グループ (Instrument Group): 未設定 (Not Set)
 - 楽器 (Instrument): 未設定 (Not Set)
- 設定状況** (Setting Status):

チャネル	楽器グループ	楽器
1	未設定	未設定
2	未設定	未設定
3	未設定	未設定
4	未設定	未設定
5	未設定	未設定
6	未設定	未設定
7	未設定	未設定
8	未設定	未設定
9	未設定	未設定
10	未設定	未設定
11	未設定	未設定
12	未設定	未設定
13	未設定	未設定
14	未設定	未設定
15	未設定	未設定

At the bottom of the dialog are three buttons: 設定 (Settings), 取消 (Cancel), and 適用 (Apply).

図 5-8 サウンド設定画面

2. 各設定項目を設定します。
各設定項目は、表 5-3 に示すとおりです。

表 5-3 サウンド設定の設定項目

設定項目	説明
再生モード選択	■再生モード 通常再生モードとビジュアルモードを選択できます。 ☐通常再生モード(※) : DirectX を使って P C 上で実際に 3 D サウンドを再生することができます。ただし、お使いの P C のサウンドカードが 3 D サウンドに対応している必要があります。 ☐ビジュアルモード : DirectX を使わない従来の再生モードです。 代替音や 3 D サウンドは鳴らせませんが、3 D サウンドを視覚的に表示する事ができます (5. 3 9. 3. 3 D サウンド表示参照)。 ■サウンドライブラリ 通常再生モードのみ有効です。 汎用サウンドライブラリと 903i 向けサウンドライブラリを選択できます。 ☐汎用サウンドライブラリ 汎用のサウンドライブラリを利用して再生を行います。 実機とは再生音が異なる事があります。また、P C 上で再生できない音色がある事があります。(代替音を設定可能です。) ☐903i 向けサウンドライブラリ 903i の実機とほぼ同等の再生が可能なサウンドライブラリを利用して再生を行います。
デバイス設定	通常再生モードのみ有効です。 P C に組み込まれているサウンドカードのサウンドデバイスを選択します。ここで選択したデバイスで再生されます。
代替音設定	通常再生モード・汎用サウンドライブラリ選択時のみ有効です。 一覧から設定したいチャンネルを選択し、「チャンネル」、「楽器グループ」、「楽器」を選択します (図 5-9)。 M F i 中の該当チャンネルで、P C 上で再生できない音色があった場合、ここで指定した音を使って再生されます。

「通常再生モード」では、サウンドライブラリが使われています。サウンドライブラリの音のタイミングなどを細かく調整したい場合は、「付録.1 エミュレーション端末の設定」を参考に skin.conf を書き換えてください。

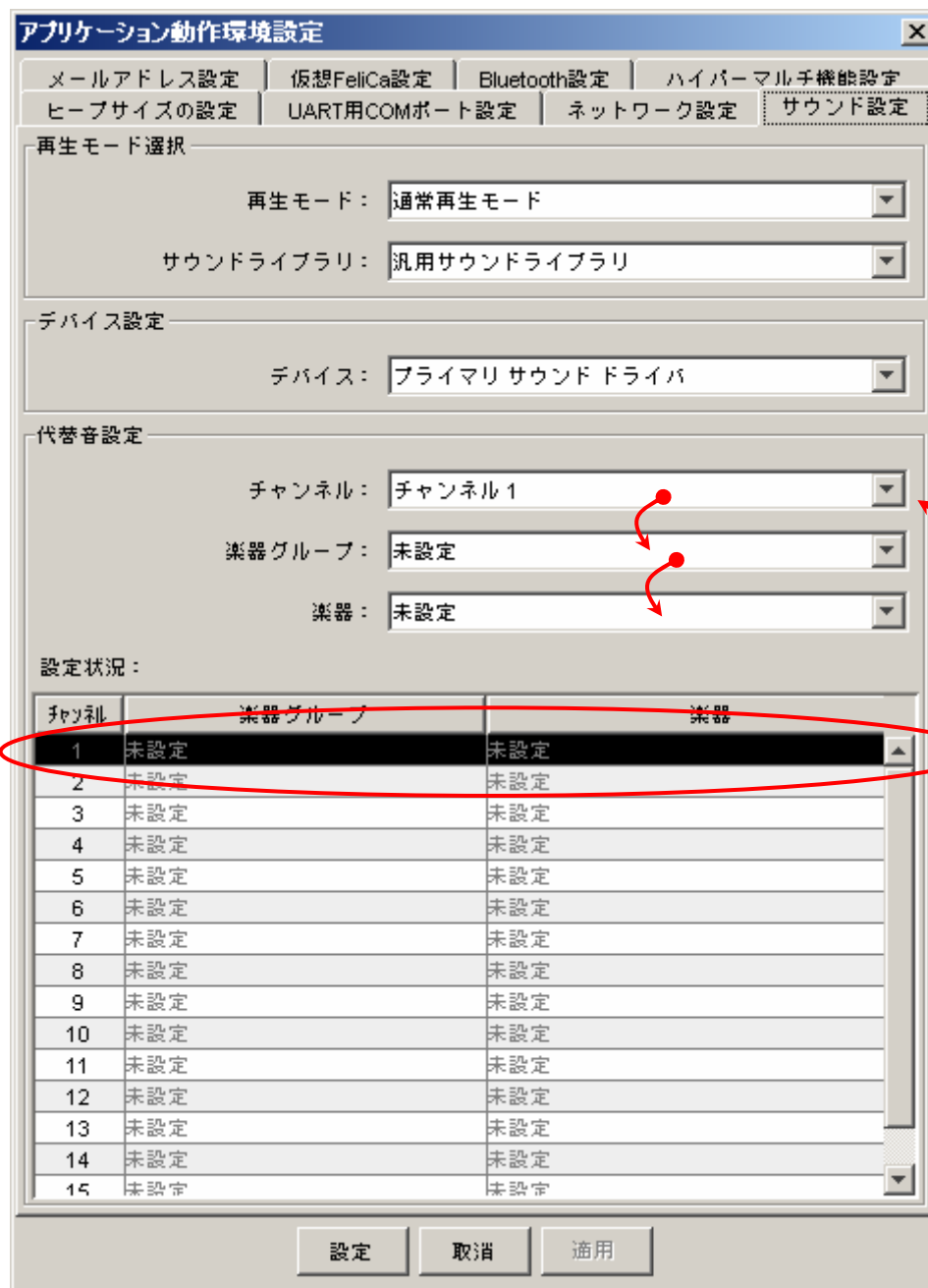


図 5-9 代替音設定

5.7. ヒープサイズの設定

ヒープサイズとは、i モード携帯電話上で動作する Java が扱えるメモリの大きさです。エミュレータでは、デフォルトで 8000Kbyte のヒープ領域が使えるように設定されています。しかし、機種の違いによって、ヒープ領域がこれよりも少ないか多いことがあります。その場合、エミュレータで動いた i アプリが i モード携帯電話で動かないことになります。そこで、次の方法により i アプリが扱えるヒープサイズを変更することができます。（実際のヒープサイズについては、各機種の説明書を参照してください。）

ヒープサイズを設定するには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「ヒープサイズの設定」を選択します（図 5-10）。

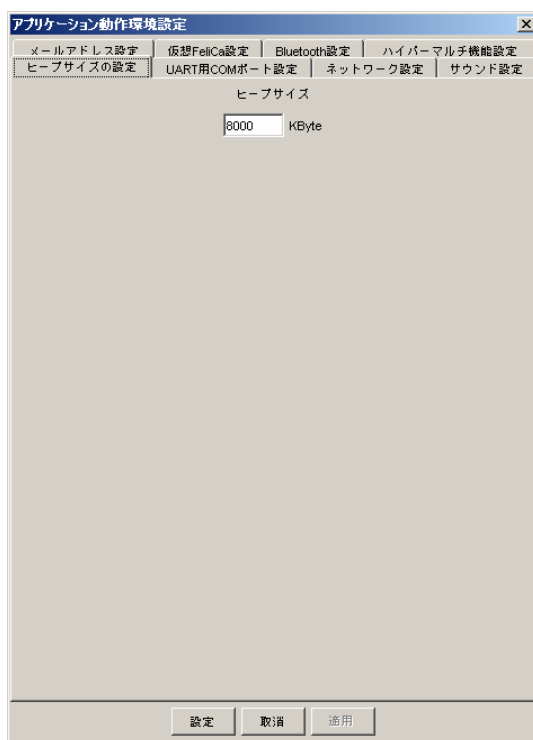


図 5-10 ヒープサイズの設定

2. 1000 ～ 32767 KByte の範囲でヒープサイズを入力して、「設定」もしくは「適用」ボタンを押します。

💡 ワンポイント

ヒープサイズは、標準の JavaVM でも使われています。
そのため、あまり小さな値を設定すると、i アプリの起動ができなかったり、動作が不安定になったりする場合があります。
とくに必要ではない限り、この値を変更することはおすすめできません。

5.8. UART用COMポートの設定

エミュレータでは、Connector クラスで UART を指定した場合に、実際に P C のシリアルポートから通信をする事ができます。

UART を用いる場合、Connector#open で以下のように指定します。

スキーム:パラメータ=値, パラメータ=値, . . .

スキーム	comm:/n	n に COMポートのポート番号を指定します。
パラメータ 1	baudrate	ボーレートを指定します。 9600, 38400, 115200
パラメータ 2	databitlen	データビット長を指定します。 8 固定
パラメータ 3	parity	パリティを以下の指定値で指定します。 NO:なし, ODD:奇数, EVEN:偶数, MARK:常に 1, SPACE:常に 0
パラメータ 4	stopbitlen	ストップビットを指定します。 1 固定
パラメータ 5	flowctrl	ハードウェアフロー制御を指定します。 ON:制御する, OFF:制御しない

例 : comm/0;baudrate=9600,databitlen=8,parity=NO,stopbitlen=1,flowctrl=OFF

※ UART の機能は携帯端末によって使えませんが、ターゲットとする携帯端末が使えるかどうか確認してください。

UART で用いるシリアル回線のCOMポートを設定するには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「UART用COMポート設定」タブを選択します（図 5-1 1）

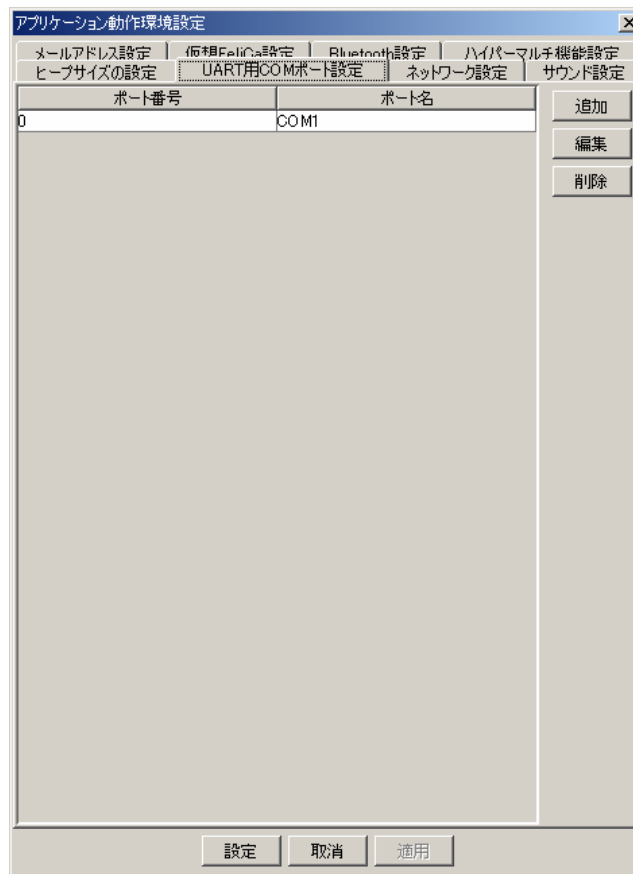


図 5-1 1 UART用COMポート設定画面

2. COMポートを追加したい場合は、「追加」を押します。
UART用COMポート編集画面が表示されるので、ポート番号とポート名を選択して「設定」を押します（図 5-1 2）。



図 5-1 2 COMポート編集画面

3. COMポートを変更したい場合は、一覧からCOMポートを選択して「編集」、削除したい場合は「削除」を押します。
4. 「設定」もしくは「適用」を押します。

5.9. Bluetooth 設定

本エミュレータでは、実際にBluetoothで通信をする事はできませんが、BluetoothのAPIを利用したiアプリで、TCP/IP上に接続された別のエミュレータとエミュレートする事ができます（図 5-13）。

※ 実際の携帯端末では、Bluetooth 搭載機のみ扱えます。

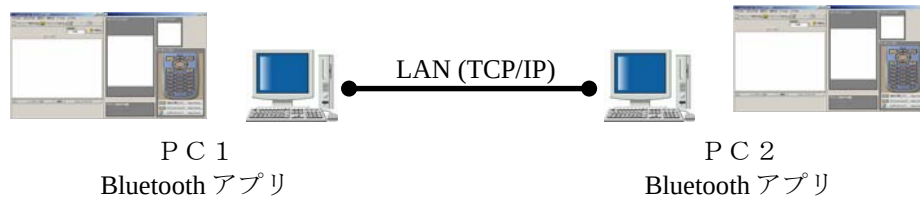


図 5-13 Bluetooth アプリのシミュレーション

Bluetooth の設定をするには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「Bluetooth設定」タブを選択します（図 5-14）

The screenshot shows the 'アプリケーション動作環境設定' (Application Environment Settings) dialog box with the 'Bluetooth設定' (Bluetooth Settings) tab selected. The dialog has several sections:

- 自Bluetoothデバイス設定** (Self Bluetooth Device Settings):
 - デバイス名称 (Device Name): DoJaBluetooth
 - Bluetoothアドレス (Bluetooth Address): 22 : 22 : 22 : 22 : 22 : 22
 - 端末属性情報 (Terminal Attribute Information): 携帯電話 (Mobile Phone)
- 登録機器** (Registered Devices):

デバイス名称	Bluetoothアドレス	端末属性情報
DoJaBluetooth	222222222222	携帯電話
DoJaBluetooth	111111111111	PC
DoJaBluetooth	112233445566	携帯電話
- 接続先設定** (Connection Destination Settings):
 - 接続先端末 (Connection Destination Terminal): 127.0.0.1

At the bottom of the dialog are three buttons: '設定' (Settings), '取消' (Cancel), and '適用' (Apply).

図 5-14 Bluetooth設定画面

2. 各設定項目を設定し、「設定」もしくは「適用」を押します。

各設定項目は表 5-4 に示すとおりです。

表 5-4 Bluetooth の設定項目

設定項目	説明	デフォルト
自 Bluetooth デバイス デバイス名	Bluetooth デバイス名称を設定します。 この情報は相手側に登録されます。	DoJaBluetooth
自 Bluetooth デバイス Bluetooth アドレス	Bluetooth アドレスを設定します。 この情報は相手側に登録されます。	112233445566
自 Bluetooth デバイス 端末属性情報	端末属性を選択します。 この情報は相手側に登録されます。	携帯電話
登録機器	Bluetooth#scan()もしくは searchAndSelectDevice() で登録された外部 機器(接続先相手)の情報が表示されます。	なし
接続先端末	i アプリが Bluetooth を使用して通信する 際の接続先 IP アドレスを設定します。 指定がない場合は、127.0.0.1 (ローカルホ スト) にコネクションを確立する試みをし ます。 なお、本設定は、Bluetooth クラスの connect / accept メソッドで意味を成しま す。	なし [省略可]

3. 登録された相手側機器情報を削除するには、削除したいデバイス名称を選択して「削除」を押します。
削除確認画面が表示されるので、「はい」を押します(図 5-15)。



図 5-15 登録機器削除確認画面

Bluetooth API で、ここに設定した登録機器が対向側（別のエミュレータ上）で表示されます（図 5-16）。

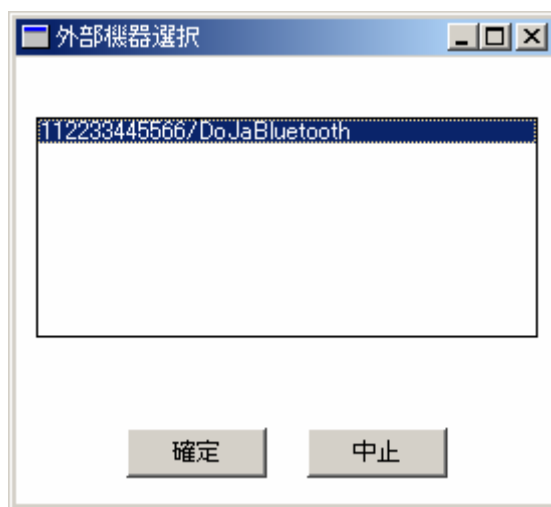


図 5-16 Bluetooth API 実行時の外部機器選択画面

5.10. 仮想 FeliCa 設定

エミュレータの仮想 FeliCa の動作を設定することにより、FeliCa の A P I を利用した i アプリで、T C P / I P 上に接続された別のエミュレータとエミュレートすることができます。

仮想 FeliCa の動作を設定するには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「仮想FeliCa設定」タブを選択します（図 5-17）

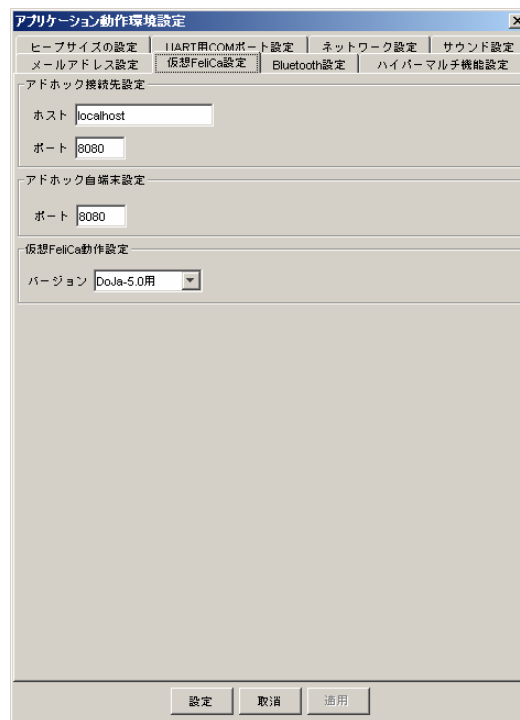


図 5-17 仮想 FeliCa 設定画面

2. 各設定項目を設定し、「設定」もしくは「適用」を押します。

各設定項目は表 5-5 に示すとおりです。

表 5-5 仮想 FeliCa の設定項目

設定項目	説明	デフォルト
アドホック 接続先設定	L A N上に存在するもう 1 台の端末と FeliCa アドホック通信をする際に接続する相手のホスト名とポート番号	localhost 8080
アドホック 自端末設定	L A N上に存在するもう 1 台の端末と FeliCa アドホック通信をする際に接続する自端末のポート番号	8080
仮想 FeliCa 動作設定	仮想 FeliCa のチップバージョンを選択します。	DoJa-5.0 用

5.11. ハイパーマルチ機能設定

エミュレータのハイパーマルチ機能でセパレートモードを設定することにより、i アプリからの端末機能の利用を制限することができます。
ハイパーマルチ対応端末においてのみ有効な機能です。

ハイパーマルチ機能を設定するには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「ハイパーマルチ機能設定」タブを選択します（図 5-18）。



図 5-18 ハイパーマルチ機能設定画面

2. 「ハイパーマルチ機能」にチェックを付けます。
3. ハイパーマルチ機能のセパレートモードを選択します。
4. 「設定」もしくは「適用」を押します。

ハイパーマルチ機能で定義されるセパレートモードを表 5-6 に示します。

表 5-6 セパレートモード定義

モード設定	用途
Aモード	基本契約番号での発信と関連データの閲覧を可能とする。
Bモード	付加番号での発信と関連データの閲覧を可能とする。
共通モード	基本契約番号、付加番号どちらも使用可能とし、全てのデータの閲覧を可能とする。

5.12. 電話帳の編集

i アプリから利用する電話帳の編集をします。

電話帳の編集をするには：

1. メニューから「設定」→「ネイティブデータ編集...」→「電話帳編集」タブを選択します（図 5-19）。

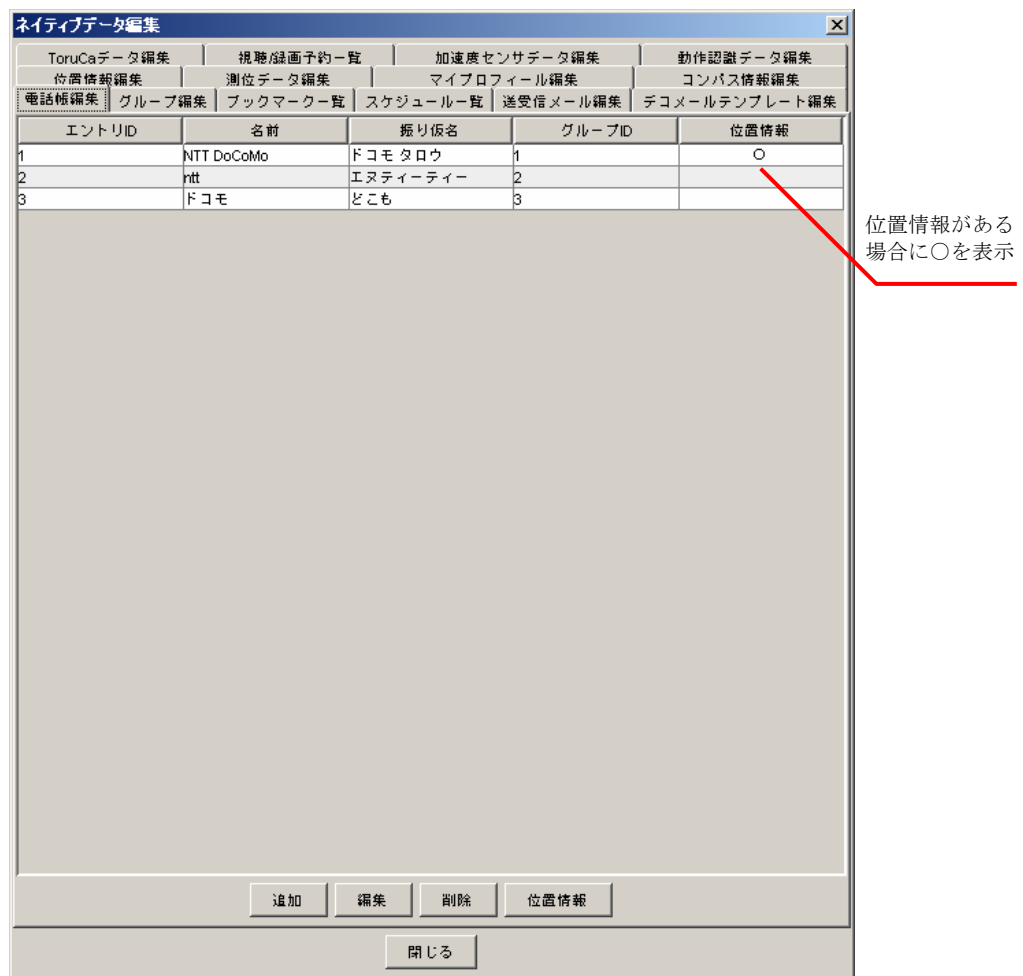


図 5-19 電話帳エントリー一覧画面

2. 電話帳エントリーを追加したい場合は、「追加」ボタンを押します。
電話帳エントリーを変更したい場合は、一覧から変更したい電話帳エントリーを選択して、「編集」ボタンを押します（図 5-20）。

図 5-20 電話帳エントリ登録画面

3. 各設定項目を設定します。

各設定項目は表 5-7 に示すとおりです。

表 5-7 電話帳エントリの設定項目

設定項目	説明	デフォルト
名前	名前。	なし
振り仮名	振り仮名。	なし
電話番号	電話番号を最大3件まで設定します。 電話番号は、ハイフン(-)を除いて指定してください。(例：09011112222)	なし [省略可]
メールアドレス	メールアドレスを最大3件まで設定します。	なし [省略可]
グループID	グループ作成時に自動的に付与されたID番号。 右の「…」ボタンを押すと、グループ一覧画面(図 5-23)で編集したグループのファイルを選択するダイアログ(図 5-21)が表示されますので、この電話帳エントリを登録したいグループのファイルを選択して「開く」を押してください。	なし [省略可]

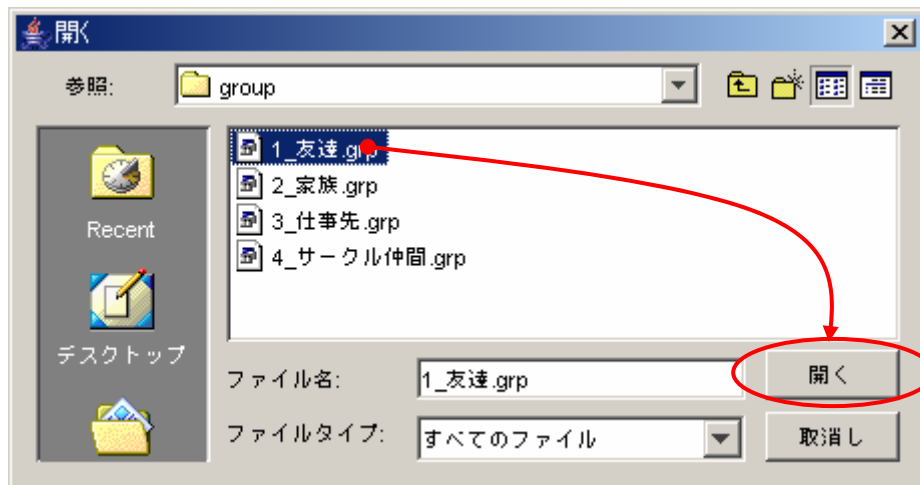


図 5-2 1 グループの選択ダイアログ

ここで作成した電話帳エントリは、
lib\phonebook\エントリ ID_名前.tel というファイルで作成されます。

4. 電話帳エントリを削除したい場合は、一覧から削除したい電話帳エントリを選択して、「削除」ボタンを押します。
5. 位置情報を登録したい場合は、一覧から登録したい電話帳エントリを選択して、「位置情報」を押します。
6. 各設定項目を設定します。
この項目は、位置情報編集から登録することもできます（「5.1 7. 位置情報編集」参照）。



図 5-2 2 位置情報登録

表 5-8 位置情報の設定項目

設定項目	説明	デフォルト
位置測位日時	G P S で測位した日時。	現在時刻
測位レベル	測位レベル。 1 ～ 3 (詳しくは、Location クラスを参照)	Unknown
緯度	緯度(N, S)をDMS単位で設定します。	なし [省略可]
経度	経度(E, W)をDMS単位で設定します。	なし [省略可]
高度	高度(m)を設定します。	なし [省略可]
測地系	測地系を選択。 選択できる測地系は、WGS84 と TOKYO です。	なし [省略可]

7. 「設定」を押すと、電話帳エントリ一覧画面の位置情報が「O」に変わります（図 5-19）。
- ここで設定した位置情報は i アプリから取得するか、位置情報電話帳起動でパラメータとして渡す事ができます（「4.2 1. 位置情報電話帳起動」参照）。

5.13. グループの編集

i アプリから利用するグループの編集をします。

グループの編集をするには：

1. メニューから「設定」→「ネイティブデータ編集...」→「グループ編集」タブを選択します（図 5-2 3）。

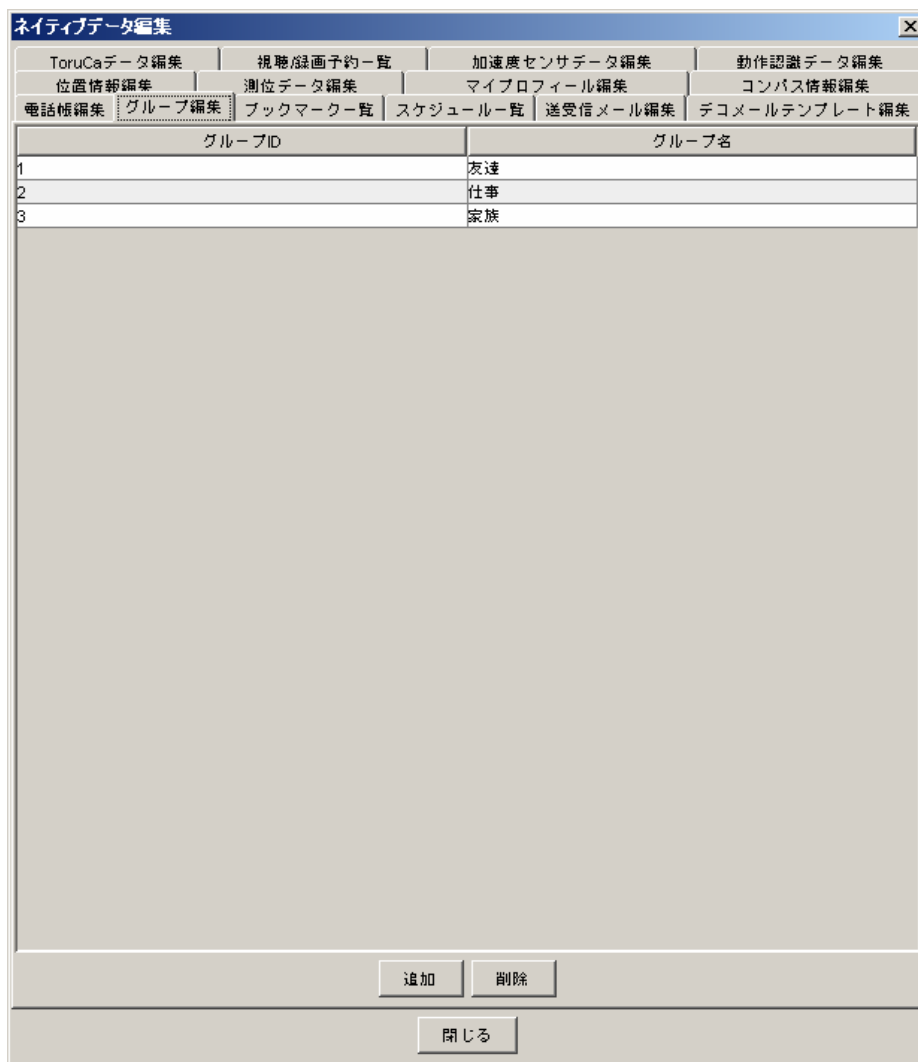


図 5-2 3 グループ一覧画面

2. グループを追加したい場合は、「追加」ボタンを押します（図 5-2 4）。



図 5-2 4 グループ追加画面

3. グループ名を入力して「設定」ボタンを押します。
ここで作成したグループは、
`lib\group\グループ ID_グループ名.grp` というファイルで作成されます。
4. グループを削除したい場合は、一覧から削除したいグループエントリを選択して、「削除」ボタンを押します。

5.14. ブックマーク一覧

i アプリから扱えるブックマークは、個人情報のセキュリティ上、設定することしかできません。

ブックマーク一覧では、アプリケーションから `Bookmark#addEntry( )` で作成されたブックマークを一覧で確認することができます。また、アプリケーションで作成したブックマークを削除することができます。

ブックマークの一覧を編集するには：

1. メニューから「設定」→「ネイティブデータ編集...」→[ブックマーク一覧] タブを選択します（図 5-25）。



図 5-25 ブックマーク一覧

2. i アプリから作成されたブックマークの一覧が表示されるので、不要なブックマークを選択して「削除」ボタンを押します。

5.15. スケジュール一覧

i アプリから扱えるスケジュールは、個人情報のセキュリティ上、設定することしかできません。

スケジュール一覧では、アプリケーションから `Schedule#addEntry( )` で設定されたスケジュールを一覧で確認することができます。また、i アプリで作成したスケジュールを削除することができます。

スケジュールの一覧を表示するには：

1. メニューから「設定」→「ネイティブデータ編集...」→「スケジュール一覧」タブを選択します（図 5-26）。

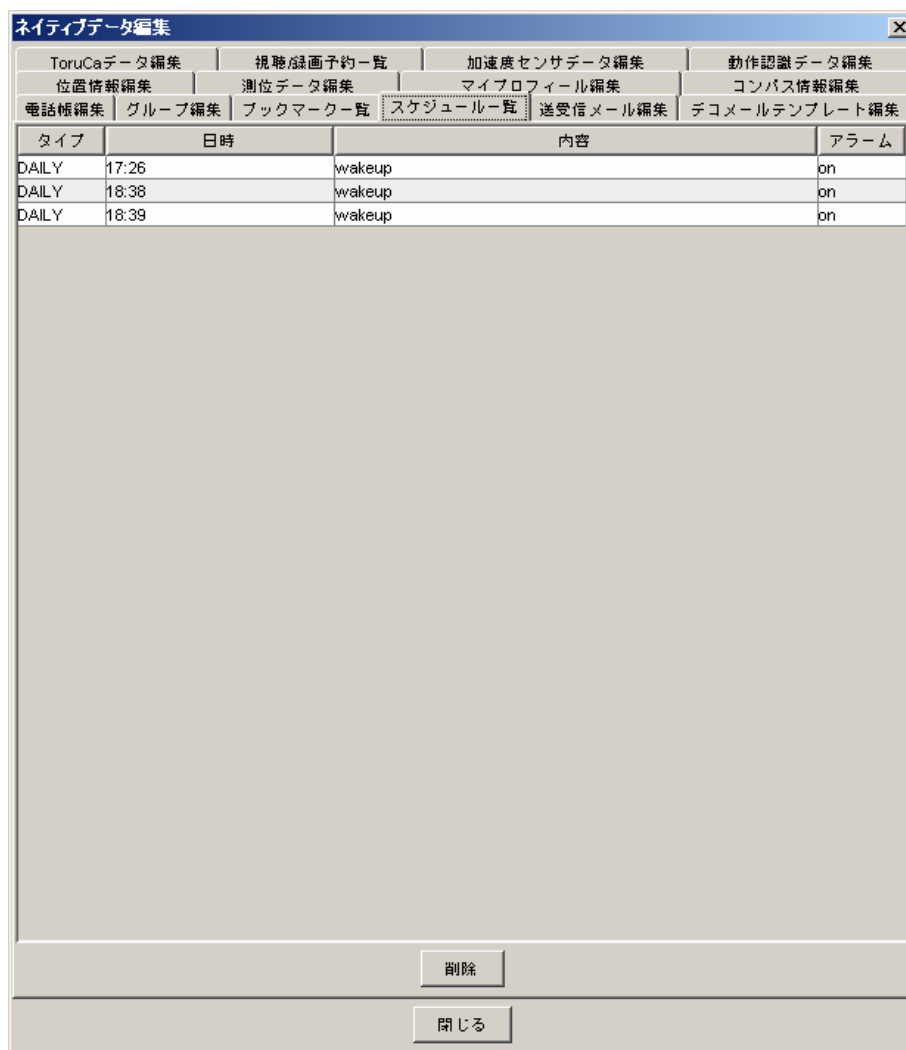


図 5-26 スケジュール一覧

2. i アプリから作成されたスケジュールの一覧が表示されるので、不要なスケジュールを選択して「削除」ボタンを押します。

5.16. 送受信メールの編集

i アプリから利用する送受信メールの編集をします。

送受信メールの編集をするには：

1. メニューから「設定」→「ネイティブデータ編集...」→[送受信メール編集] タブを選択します（図 5-27）。

状態	受信日	From	タイトル
既	2007/01/26 00:00	message_app_user1@docomo.ne.jp	おめでとう
既	2007/01/01 11:11	message_app_user2@docomo.ne.jp	ありがとう
既	2006/05/01 12:12	message_app_user3@docomo.ne.jp	どうぞ

図 5-27 送受信メール一覧画面

2. 編集したいメールボックスを選択すると、一覧が切り替わります（図 5-27）。メールボックスの種類は以下の3つから選べます。

- 受信ボックス
受信したメールを保存します。
- 送信ボックス
送信したメールを保存します。
- 未送信ボックス
送信前（編集途中）、または、送信に失敗したメールを保存します。

3. メールを追加したい場合は、「追加」ボタンを押します。
メールを変更したい場合は、一覧から変更したいメールを選択して、「編集」ボタンを押します（図 5-28、図 5-29）。

メール編集(受信ボックス)

To: message_app_usr0@docomo.ne.jp,

From: message_app_usr1@docomo.ne.jp

タイトル: おめでとう

受信日: 2005 年 06 月 30 日 8 時 15 分

MessageCode: 0000000000

ネイティブメール: ☐

状態: ☒ 既読 ☐ 未読

本文

お誕生日おめでとう

設定 取消

図 5-28 メール編集 (受信ボックス) 画面

メール編集(送信ボックス)

To: message_app_usr1@docomo.ne.jp

From: message_app_usr0@docomo.ne.jp

タイトル: Re:おめでとう

送信日: 2005 年 06 月 30 日 8 時 20 分

MessageCode: 0000000000

ネイティブメール: ☐

宛先状態: ☒ XString ☐ 通常文字列

本文

ありがとう

設定 取消

図 5-29 メール編集 (送信ボックス) 画面

4. 各設定項目を入力して「設定」を押します。

各設定項目は表 5-9 に示すとおりです。

表 5-9 送受信メールの設定項目

設定項目	説明	デフォルト
T o	受信ボックスの場合は受信者、送信ボックスおよび未送信ボックスの場合は受信者のアドレス。 複数指定する場合は、カンマ(,)で区切って指定します。	なし
F r o m	受信ボックスの場合は送信者、送信ボックスおよび未送信ボックスの場合は送信者のアドレス。 複数指定する場合は、カンマ(,)で区切って指定します。	なし
タイトル	メールのタイトル。	なし [省略可]
受信日	メールを受信した日時。	なし
送信日	メールを送信した日時。	なし
MessageCode	A D F に設定した MessageCode。	なし
ネイティブメール	ネイティブメール（端末が管理するメール＝Mail）の場合は、チェックをつけます。チェックをつけた場合、このメールは、メッセージiアプリのメールボックスに移動されたネイティブメールとなります。 ネイティブメールには、バウンダリ（--B:P以降）は付与されません。	チェックなし (Message)
状態	既読か未読の状態。 既に読まれたメールは「既読」、まだ読まれていないメールは「未読」にします。	既読
宛先状態	宛先の完全カプセル化文字列（A P I から取得できない文字列）の指定。 宛先を完全カプセル化文字列にする場合は XString にします。 電話帳エントリから取得したメールアドレスを指定した場合、および、受信ボックスから返信するメールの場合は、XString になります。	XString
本文	メールの本文。	なし [省略可]

ここで作成したメールは、それぞれ以下のファイルで作成されます。

受信ボックス： lib\mail\received\メッセージコード\受信日_通番.mail

送信ボックス： lib\mail\sent\メッセージコード\送信日_通番.mail

未送信ボックス： lib\mail\unsent\メッセージコード\送信日_通番.mail

5. メールを削除したい場合は、一覧から削除したいメールを選択して、「削除」ボタンを押します。
削除してもよいか確認画面が表示されるので「はい」を選択すると削除されます（図 5-30）。

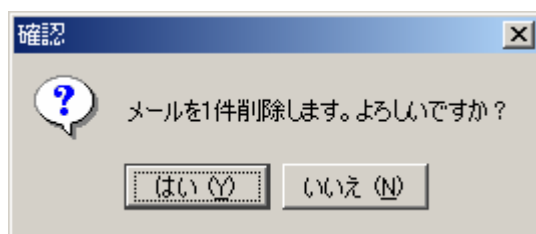


図 5-30 メール削除確認画面

メールファイルの形式については、「10.6. メールファイル」を参照してください。

また、メッセージコードは、ADFの MessageCode の値ですが、このキー項目が設定されていない場合は、“0000000000”であるとみなしてメールボックスとメールを作成します。そのため、ADFの MessageCode を設定する前に作成したメールをメッセージ i アプリで使いたい場合には、ADFの MessageCode を“0000000000”に設定してください。（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」を参照）

メール中には、i アプリ固有のバイナリデータが付与されます。
メールに付与されるバイナリデータは、以下のファイルに保存されています。

バウンダリ部のバイナリデータ : lib\mail\boundary.data

バイナリデータを変更したい場合は、このファイルの内容を変更してください。
ただし、このバイナリデータは、Base64 でエンコードされた値でなくてはなりません。

また、各メールの状態は、以下のファイルに保存されています。

状態 : lib\mail\receive\メッセージコード\mail.state
宛先状態 : lib\mail\sent\メッセージコード\xstring.state
lib\mail\unsent\メッセージコード\xstring.state
メールアドレス : lib\mail\myAddress

これらのファイルは、特に修正する必要はありません。

■ 作成したメールのメッセージIDは、1～999 で付与されます。

5.16.1. ショートメールの扱いについて

ショートメールをメールボックスに作成する場合は、メールの追加で、それぞれ次のように指定します。（図 5-3 1、図 5-3 2）

- 受信ボックス
From : iMS
Subject : 電話番号(ショートメール)
ネイティブメール
- 送信ボックス
To : iMS
Subject : 電話番号(ショートメール)
ネイティブメール
- 未送信ボックス
To : iMS
Subject : 電話番号(ショートメール)
ネイティブメール

図 5-3 1 受信ボックスのショートメール作成

図 5-3 2 送信ボックス／未送信ボックスのショートメール作成

5.16.2. メールの送受信エミュレート機能

メールを扱う i アプリは、メールの送受信をエミュレートする事ができます。

メールの送受信をエミュレートするには：

1. メールアドレスの設定画面に送受信するメールアドレスを設定しておきます。
(「5.3. メールアドレスの設定」を参照)
2. エミュレートする i アプリの MessageCode を同一の値にします。
(「4.7. アプリケーションディスクリプタファイル (ADF) の編集」を参照)
※ 異なる MessageCode (i アプリ) 同士でのメールはやり取りできません。
3. i アプリからメールを送信します。
(MessageAgent#send() を実行します。)
4. 送信したメールが、宛先に指定したメールアドレスに該当する受信ボックスに受信メールで格納されます。
ただし、1. で宛先に指定したメールアドレスを設定した i アプリが存在しない、もしくは、MessageCodeが異なる場合、受信メールは作成されません (図 5-33)。

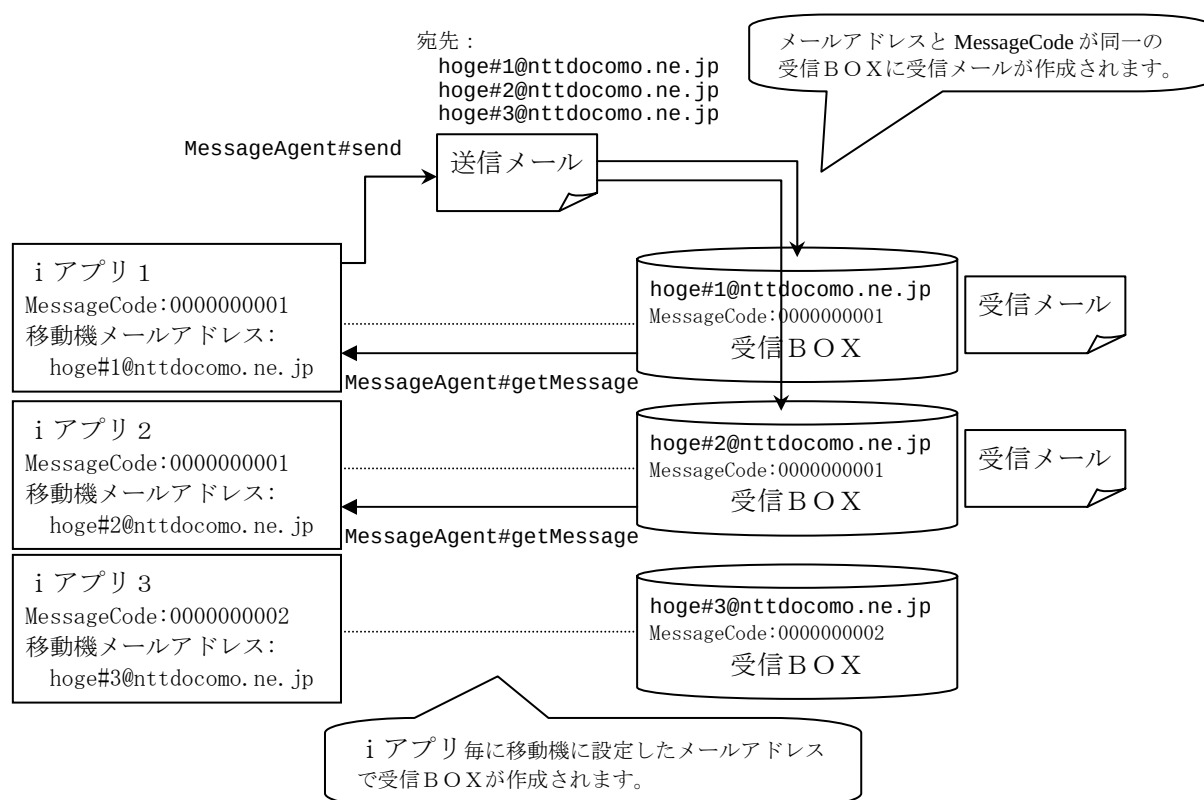


図 5-33 i アプリによるメールの送受信

※ 同一PC上でエミュレータを複数起動しても同じ環境 (同じメールアドレス) になりますので、ここでは、メールアドレスを変更して、それぞれのアプリを順次、実行した場合を指しています。

5.17. 位置情報編集

i アプリから利用する位置情報（GPSの測位結果）の編集をします。

位置情報を編集するには：

1. メニューから「設定」→「ネイティブデータ編集...」→「位置情報編集」タブを選択します（図 5-3 4）。

ネイティブデータ編集	
電話帳編集	グループ編集
TrueCallデータ編集	視聴録画予約一覧
位置情報編集	測位データ編集
	マイプロフィール編集
	コンパス情報編集

エントリID	位置測位日時
1	2006/12/18 14:22:44.609
2	2007/02/02 10:58:03.812
3	2007/02/02 10:56:05.500

追加 編集 削除 すべて削除 画像登録 電話帳登録

閉じる

図 5-3 4 位置情報一覧画面

2. 位置情報を追加したい場合は、「追加」ボタンを押します。
位置情報を変更したい場合は、一覧から変更したい位置情報を選択して、「編集」ボタンを押します（図 5-3 5）。
ここで登録した位置情報で位置情報起動ができます（「4.19. 位置情報起動」参照）。また、i アプリで取得することもできます。

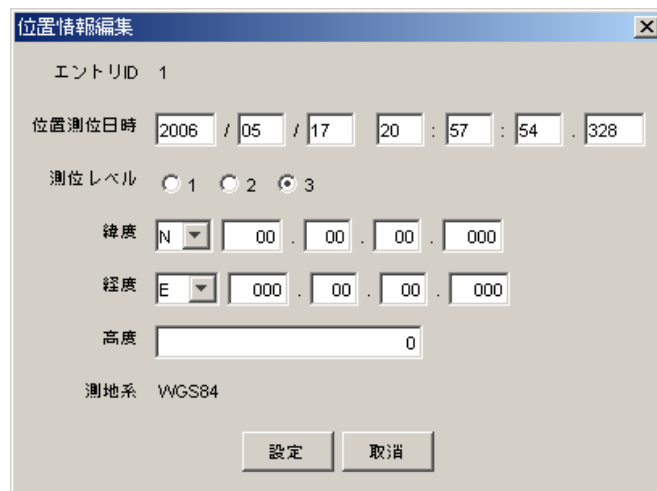


図 5-3 5 位置情報編集画面

3. 位置情報を削除したい場合は、一覧から削除したい位置情報を選択して、「削除」ボタンを押します。
全ての位置情報を削除したい場合は、「すべて削除」ボタンを押します。
削除してもよいか確認画面が表示されるので「はい」を選択すると削除されます（図 5-3 6）。

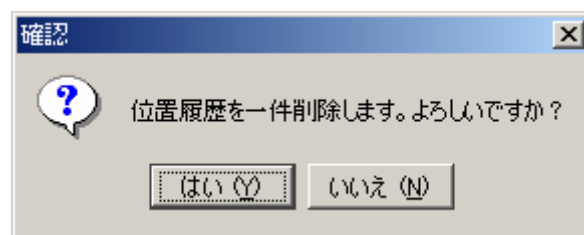


図 5-3 6 位置情報削除確認画面

4. 位置情報を画像に設定したい場合は、一覧から設定したい位置情報を選択して、「画像登録」ボタンを押します。
エン트리IDの設定画面が表示されるので、画像のエン트리IDを指定して「設定」を押します（図 5-3 7）。
ここで設定したエン트리IDで位置情報画像起動ができます（「4. 2 0. 位置情報画像起動」参照）。

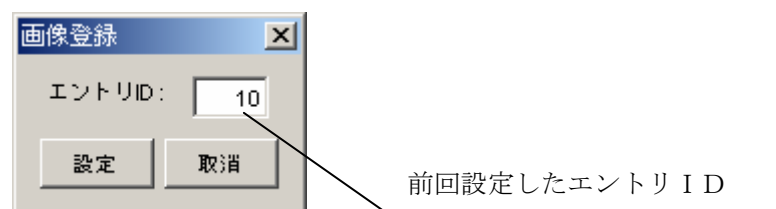


図 5-3 7 位置情報画像エン트리ID設定画面

5. 位置情報を電話帳に設定したい場合は、一覧から設定したい位置情報を選択して、「電話帳登録」ボタンを押します。
電話帳のエントリIDの設定画面が表示されるので、電話帳のエントリIDを指定して「設定」を押します（図 5-38）。
ここで設定した電話帳エントリの位置情報が書き変わります（「5.12. 電話帳の編集」、「4.21. 位置情報電話帳起動」参照）。

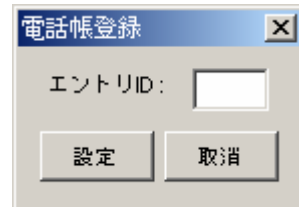


図 5-38 位置情報電話帳エントリID設定画面

5.18. 測位データ編集

i アプリから利用するGPS測位データを編集します。

GPSを用いたi アプリを作成する際にGPSで測位されたデータを登録しておきます。
これによりGPSを用いたi アプリからこの測位データが取得できるようになります。

測位データの編集をするには、

1. メニューから「設定」→「ネイティブデータ編集...」→「測位データ編集」タブを選択します（図 5-39）。

緯度	経度	高度	測位レベル	測位時間	測地系	測位方式
N0.000000	E0.000000	0	3	5	WGS84	GPS測位
N0.000000	E0.000000	0	3	5	WGS84	GPS測位
N0.000000	E0.000000	0	3	5	WGS84	GPS測位

図 5-39 測位データ一覧画面

2. 測位データを追加したい場合は、「追加」ボタンを押します。
測位データを変更したい場合は、一覧から変更したい測位データを選択して、「編集」ボタンを押します（図 5-40）。

測位データ編集

測位レベル ☐ 1 ☐ 2 ☒ 3

緯度 N 0 . 000000

経度 E 0 . 000000

高度 0 m

測位時間 30 秒

測地系 WGS84

測位方式 GPS測位

設定 取消

図 5-4 0 測位データ編集画面

3. 各設定項目を設定して「設定」を押します。

各設定項目は表 5-1 0 に示すとおりです。

表 5-1 0 測位データの設定項目

設定項目	説明	デフォルト
測位レベル	測位レベル（誤差）を1～3で設定します。 レベル1の最悪値：2147483647 [m] レベル2の最悪値：299 [m] レベル3の最悪値：49 [m]	1
緯度	DEGREE単位で、N:北緯 S:南緯 と 角度を小数で指定します。	N60.000000
経度	DEGREE単位で、E:東経 W:西経 と 角度を小数で指定します。	E60.000000
高度	高度を m で指定します。	0
測位時間	測位が完了するまでの時間を秒単位で指定します。 ここで設定した測位時間が、 LocationProvider#getLocation で指定したタイムアウト値よりも大きい場合、測位に失敗します。	30

測地系と測位方式は固定です。

4. 測位データを削除したい場合は、一覧から削除したい測位データを選択して、「削除」ボタンを押します。
削除してもよいか確認画面が表示されるので「はい」を選択すると削除されます（図 5-4 1）。

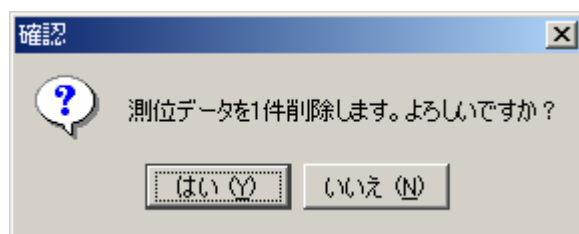


図 5-4 1 測位データ削除確認画面

5. 測位データの順序を変更する場合は、順序を変更したい測位データを選択して、「上へ」もしくは「下へ」を押します。
6. 「インポート」で、CSV形式の測位データを取り込みます。
「インポート」を押下すると、ファイル名を入力する画面が表示されるので、CSV形式の測位データが格納されたファイルを指定して「実行」を押します。
※詳細な手順については後述の【インポート機能について】を参照してください。
7. 「エクスポート」で、編集した測位データをCSV形式のファイルに保存します。
「エクスポート」を押下すると、ファイル名を入力する画面が表示されるので、測位データを保存したいファイルを指定して「実行」を押します。
※詳細な手順については後述の【エクスポート機能について】を参照してください。

【インポート機能について】

1. 「インポート」を押下すると、ファイル名を入力する画面が表示されます（図 5-4 2）。

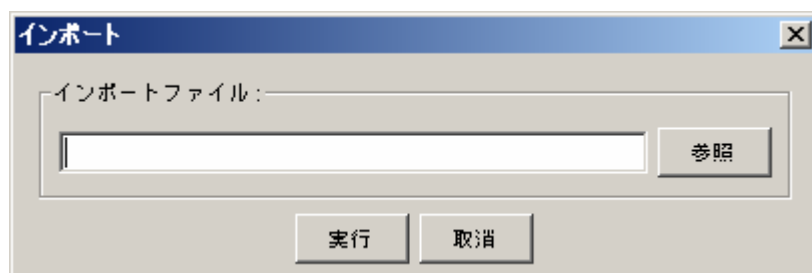


図 5-4 2 インポート画面

2. CSV形式でデータが格納されたファイル名を直接入力するか、もしくは「参照」を押下後ファイルを指定して「実行」を押します。

3. インポート処理中の画面が表示され（図 5-4 3）、100%になるとデータのインポート処理は完了です。処理中に「中断」を押す事で処理の中断も可能です。

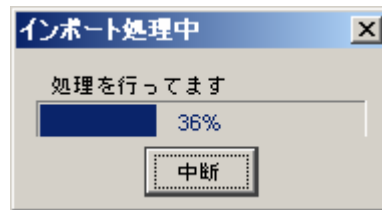


図 5-4 3 インポート処理中画面

4. 入力されたCSVファイルにデータ不正があった場合、インポート処理中に以下の警告画面が表示されます（図 5-4 4）。
「中断」を押下した場合、既に取り込んでいるデータを反映して処理を終了します。
「続行」を押下した場合、そのデータを無視して処理を継続します。
「取消」を押下した場合、既に取り込んでいるデータを破棄して処理を終了します。

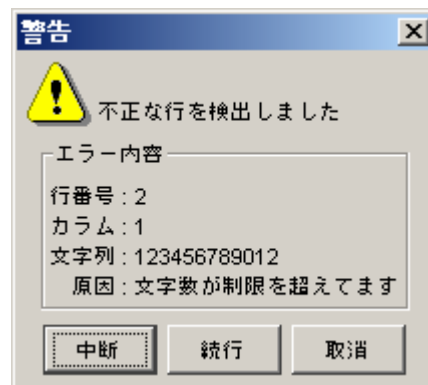


図 5-4 4 入力データ警告画面

【エクスポート機能について】

1. 「エクスポート」を押下すると、ファイル名を入力する画面が表示されます（図 5-4 5）。

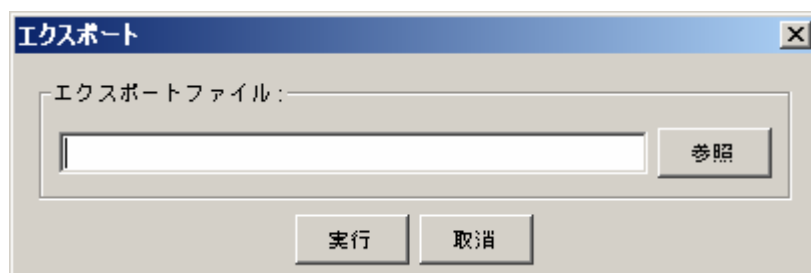


図 5-4 5 エクスポート画面

2. CSV形式のデータを格納するファイル名を直接入力するか、もしくは「参照」を押下後ファイルを指定して「実行」を押します。

3. エクスポート処理中の画面が表示され（図 5-4 6）、100%になるとデータのエクスポート処理は完了です。処理中に「中断」を押す事で処理の中断も可能です。

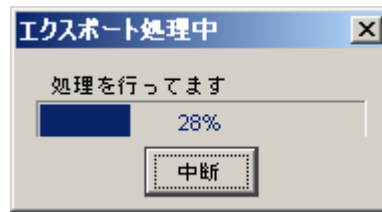


図 5-4 6 エクスポート処理中画面

4. エクスポートするファイルが既に存在した場合、以下の選択画面が表示されます（図 5-4 7）。
「上書き」を押下した場合、既に存在するファイルの内容に対し上書きする形でデータが出力されます。
「追加」を押下した場合、既に存在するファイルの内容に対し追記する形でデータが出力されます。
「取消」を押下した場合、処理を終了します。

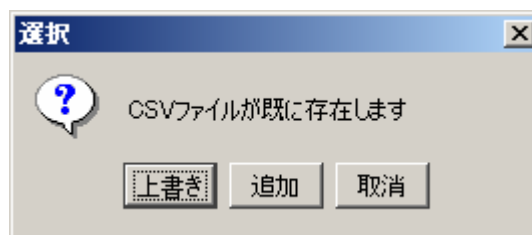


図 5-4 7 エクスポート処理選択画面

5.19. コンパス情報編集

i アプリから利用する電子コンパス情報を編集します。

コンパス情報の編集をするには：

1. メニューから「設定」→「ネイティブデータ編集...」→「コンパス情報編集」タブを選択します（図 5-48）。

The screenshot shows a dialog box titled 'ネイティブデータ編集' (Native Data Edit). It has a tabbed interface with the following tabs: 電話帳編集, グループ編集, ブックマーク一覧, スケジュール一覧, 送受信メール編集, デコメールテンプレート編集, ToruCaデータ編集, 視聴/録画予約一覧, 加速度センサーデータ編集, 動作認識データ編集, 位置情報編集, 測位データ編集, マイプロフィール編集, and コンパス情報編集 (which is the active tab). Inside the dialog, there is a label '磁北からの方位角' (Magnetic North Azimuth) followed by two input fields containing '000' and '000000'. At the bottom of the dialog, there are two buttons: '設定' (Settings) and '閉じる' (Close).

図 5-48 コンパス情報編集画面

2. 磁北からの方位角に `Compass#getAzimuth()` で取得できる方位角を **DEGREE 単位**で指定して「設定」を押します。

5.20. マイプロフィール編集

i アプリから利用するマイプロフィールを編集します。

マイプロフィールの編集をするには：

1. メニューから「設定」→「ネイティブデータ編集...」→「マイプロフィール編集」タブを選択します（図 5-4 9）。

図 5-4 9 マイプロフィール編集画面

2. 各設定項目を入力して「設定」を押します。
各設定項目は表 5-1 1 に示すとおりです。

表 5-1 1 マイプロフィールの設定項目

設定項目	説明	デフォルト
名前	名前を指定します。	なし
名前カナ	仮名を指定します。	なし
電話番号	電話番号 1、2 を指定します。	なし
メールアドレス	メールアドレス 1、2 を指定します。	なし
住所	住所を指定します。	なし
生年月日	生年月日を指定します。	なし
マイプロフィール選択項目	i アプリに渡してもよい情報にチェックを付けます。 実際の携帯端末では DoJa API の実行時にマイプロフィール選択画面が表示され、ユーザが選択します。	全て

ここで設定した情報は、OwnerProfile で取得できます。

5.21. トルカデータ編集

i アプリから利用するトルカデータを編集します。

トルカのデータを編集するには：

1. メニューから「設定」→「ネイティブデータ編集...」→「ToruCaデータ編集」タブを選択します（図 5-50）。

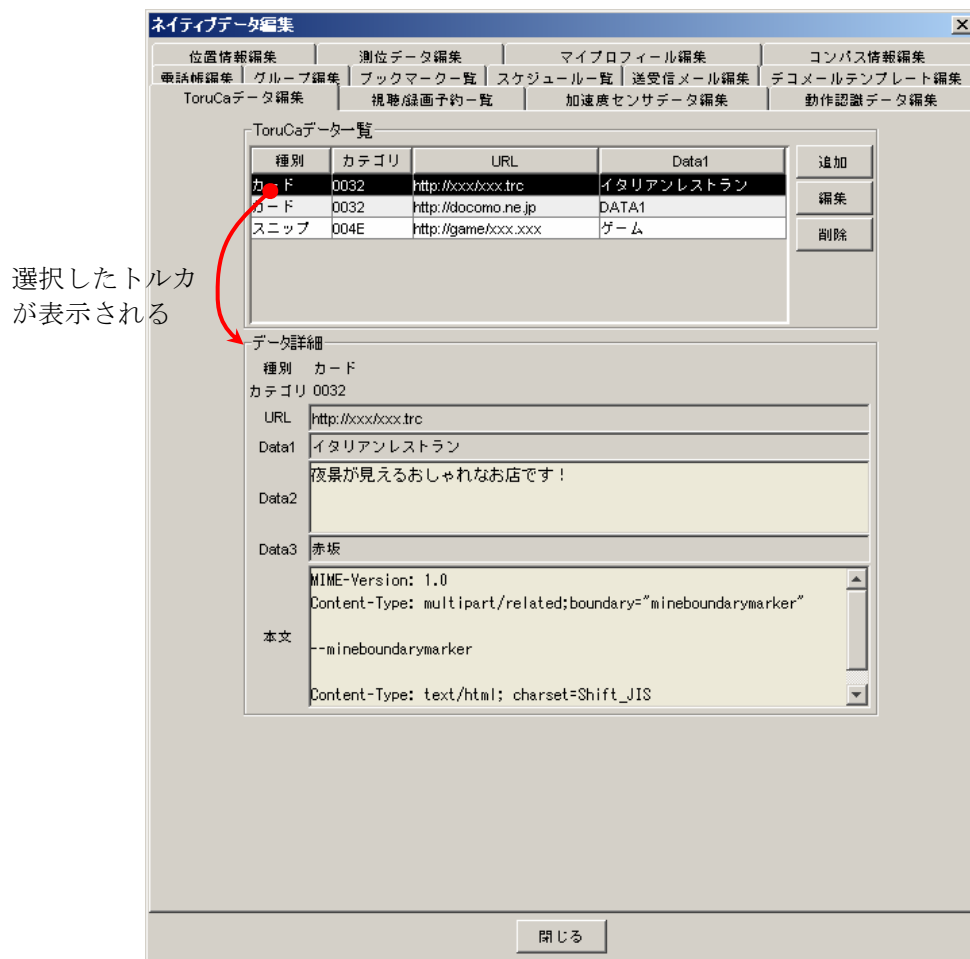


図 5-50 トルカデータ一覧画面

2. ToruCa データ一覧からトルカを選択すると、そのトルカの内容が表示されます。
3. トルカを追加したい場合は、「追加」ボタンを押します。
トルカを変更したい場合は、一覧から変更したいトルカを選択して、「編集」ボタンを押します（図 5-51）。
ここで設定した値は、i アプリで取得することができます。

ToruCa編集画面

データ詳細

種別: カード

ジャンル: ぐるめ

コード: 0032

バージョン: 2.0

IP-ID: 12345678

カナID: ポー

再配布識別子: 1

色ID: 1

有効期限: 051225

URL: http://xxx/xxx.trc

Data1: イタリアンレストラン ポーノポーノ
夜景が見えるお洒落なお店です!

Data2:

Data3: 赤坂

本文: MIME-Version: 1.0
Content-Type: multipart/related; boundary=
--mimeboundarymarker
Content-Type: text/html; charset=Shift_J
Content-Transfer-Encoding: 8bit

設定 取消

図 5-5 1 トルカデータ編集画面

4. 各設定項目を入力して「設定」ボタンを押します。

各設定項目は表 5-1 2 に示すとおりです。

表 5-1 2 トルカデータの設定項目

設定項目	説明	デフォルト
種別	スニップかカードかを選択します。	スニップ
ジャンル	トルカのジャンルを選択します。 実際の携帯端末では該当するカテゴリアイコンが表示されます。	なし [省略可]
コード	そのジャンルに該当するカテゴリコードを選択します。	0000 [省略可]
バージョン	バージョンを指定します。	なし [省略可]

IP-ID	IP-ID を指定します。	なし [省略可]
カナ ID	カナ I D を指定します。全角 2 文字	なし [省略可]
再配布識別子	再配布識別子を指定します。	なし [省略可]
色 ID	色 I D を指定します。0 ～ 9	なし [省略可]
有効期限	有効期限を指定します。YYMMDD 種別でスニップを選択した場合は、指定できません。	なし [省略可]
U R L	U R L を指定します。 U R L は実際の携帯端末上では表示されません。	なし [省略可]
Data1	トルカデータ 1 を指定します。	なし [省略可]
Data2	トルカデータ 2 を指定します。	なし [省略可]
Data3	トルカデータ 3 を指定します。	なし [省略可]
本文	トルカカード本体を指定します。 種別でスニップを選択した場合は、指定できません。	なし [省略可]

5.22. デコメールテンプレート編集

i アプリから利用するデコメールテンプレートを編集します。

デコメールテンプレートを編集するには:

1. メニューから「設定」→「ネイティブデータ編集...」→「デコメールテンプレート編集」タブを選択します（図 5-5 2）。

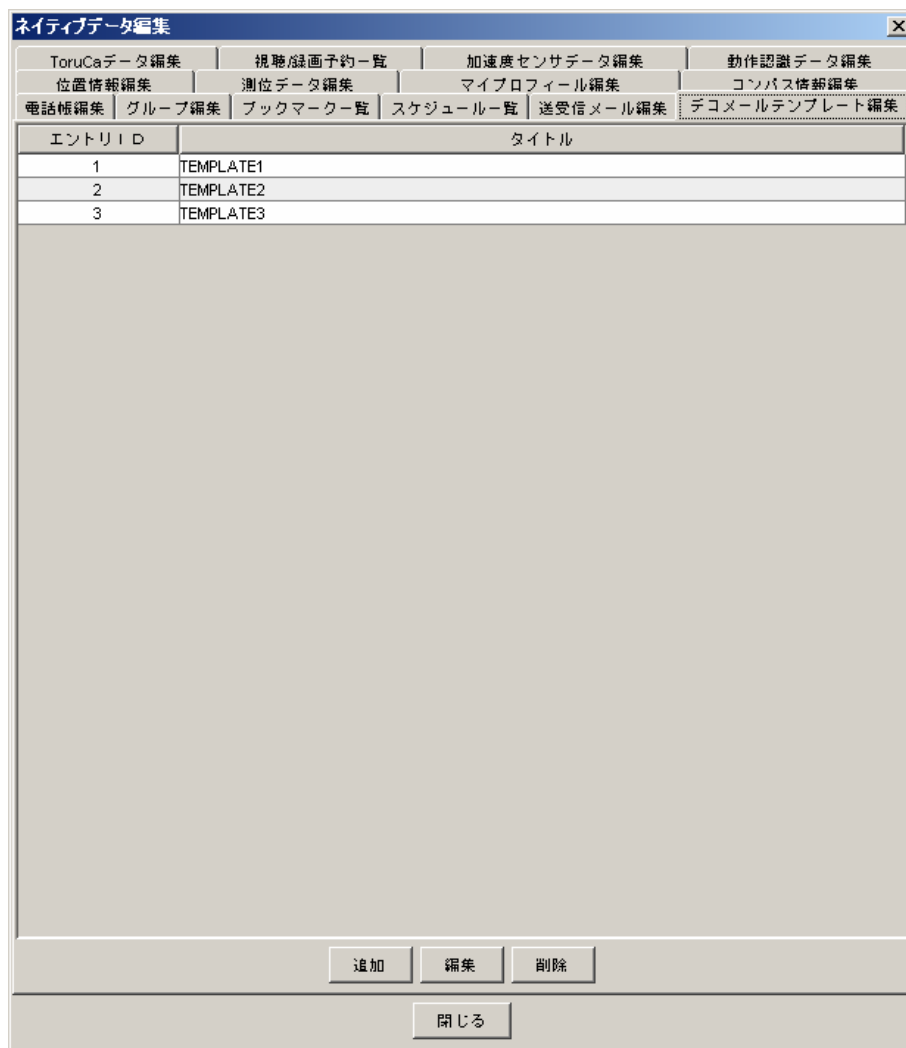


図 5-5 2 デコメールテンプレート一覧画面

2. デコメールテンプレートを追加したい場合は、「追加」ボタンを押します。デコメールテンプレートを変更したい場合は、一覧から変更したいデコメールテンプレートを選択して、「編集」ボタンを押します（図 5-5 3）。
ここで登録したデコメールテンプレートを i アプリから取得することができます。

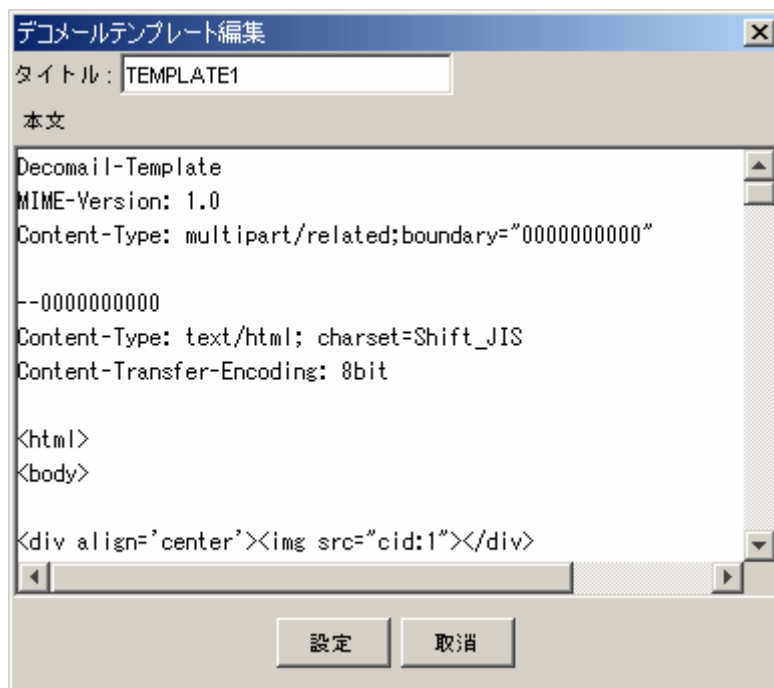


図 5-5 3 デコメールテンプレート編集画面

3. 各設定項目を入力して「設定」ボタンを押します。

各設定項目は表 5-1 3 に示すとおりです。

表 5-1 3 デコメールテンプレートの設定項目

設定項目	説明	デフォルト
タイトル	デコメールテンプレートのタイトルを指定します。 指定しなかった場合は、TEMPLATE _n になります。(n : エントリ ID)	TEMPLATE _n [省略可]
本文	テンプレート本文を指定します。 ここで指定するデコメールテンプレートは形式が正しくないとなりません。 <例> Decomail-Template MIME-Version: 1.0 Content-Type: multipart/related;boundary="0000000000" --0000000000 Content-Type: text/html; charset=Shift_JIS Content-Transfer-Encoding: 8bit <html><body>サンプル</body></html> --0000000000	なし

※ デコメールテンプレートは登録した i アプリからしか取得できませんが、エミュレータでは登録した i アプリかどうかのチェックはおこないません。
また、デコメールテンプレートのサイズもチェックしていません。
(getRemainingBytes(DecomailDraft mail) は常に 10000 を返しています。)

5.23. 視聴/録画予約一覧

i アプリから扱える視聴/録画予約情報は、個人情報のセキュリティ上、設定することしかできません。

視聴/録画予約一覧では、アプリケーションから `DTVSchedule#addEntry()` で作成された視聴/録画予約情報を一覧で確認することができます。また、アプリケーションで作成した視聴/録画予約情報を削除することができます。

情報の一覧を編集するには：

1. メニューから「設定」→「ネイティブデータ編集...」→「視聴/録画予約一覧」タブを選択します（図 5-5 4）。

選択した視聴/
録画予約情報
が表示される

ネイティブデータ編集

視聴	開始時刻	終了時刻	チャンネル名	番組名
録画	2006/12/12(Tue) 18:04	2006/12/14(Thu) 18:04	サッカー	テクニク
録画	2006/12/12(Tue) 18:04	2006/12/14(Thu) 18:04	サッカー	soccer
録画	2006/12/12(Tue) 18:04	2006/12/14(Thu) 18:04	サッカー	試合

詳細情報

視聴/録画	録画	サービス識別	1
開始時刻	2006/12/12(Tue) 18:04	物理チャンネル	13
終了時刻	2006/12/14(Thu) 18:04	系列識別	1
チャンネル名	サッカー	繰返し	1回限り
番組名	テクニク		

削除 閉じる

図 5-5 4 視聴/録画予約一覧

- ※ 繰返し種別が一回限り (ONE_TIME) 以外の視聴/録画予約情報の開始時刻・終了時刻につきましては、繰返し種別に該当する日時のうち、エミュレータを実行している PC の現在時刻の直近の日時が表示されます。

2. 視聴/録画予約一覧から視聴/録画予約情報を選択すると、その視聴/録画予約情報の内容が表示されます。
3. 視聴/録画予約情報を削除したい場合は、一覧から削除したい視聴/録画予約情報を選択して、「削除」ボタンを押します。

5.24. 加速度センサデータ編集

i アプリから利用する加速度センサのデータを編集します。

加速度センサのデータを編集するには：

1. メニューから「設定」→「ネイティブデータ編集...」→[加速度センサデータ編集] タブを選択します（図 5-5 5）。

X軸方向	Y軸方向	Z軸方向	ロール角	ピッチ角	向き
10	15	5	60	40	90
10	20	10	50	30	90
15	15	5	40	40	90

図 5-5 5 加速度センサデータ編集

2. 加速度センサデータを追加したい場合は、「追加」ボタンを押します。
 加速度センサデータを変更したい場合は、一覧から変更したい加速度センサデータを選択して、「編集」ボタンを押します（図 5-5 6）。

加速度センサデータ設定

加速度

X軸方向 (mg)

Y軸方向 (mg)

Z軸方向 (mg)

端末の傾き

ロール角 (度)

ピッチ角 (度)

メインスクリーンの縦横の向き

向き (度)

設定 取消

図 5-5 6 加速度センサデータ編集画面

3. 各設定項目を設定して「設定」を押します。

各設定項目は表 5-1 4 に示すとおりです。

表 5-1 4 加速度センサデータの設定項目

設定項目	説明	デフォルト
加速度	X 軸方向：携帯電話の左から右方向 Y 軸方向：携帯電話の下から上方向 Z 軸方向：携帯電話の背面から前面方向	0
端末の傾き	ロール角：-180～179 ピッチ角：-180～179	0
メインスクリーンの縦横の向き	スクリーンの縦横の向きを 0 度、90 度、180 度、270 度、-1(異常の場合) から選択します。	0

4. 加速度センサデータを削除したい場合は、一覧から削除したい加速度センサデータを選択して、「削除」ボタンを押します。
削除してもよいか確認画面が表示されるので「はい」を選択すると削除されます（図 5-57）。

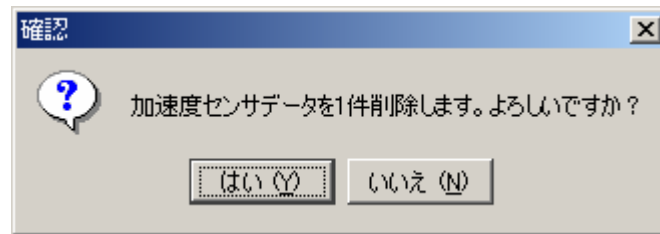


図 5-57 加速度センサデータ削除確認画面

5. 加速度センサデータの順序を変更する場合は、順序を変更したい加速度センサデータを選択して、「上へ」もしくは「下へ」を押します。
6. 「インポート」で、CSV形式の加速度センサデータを取り込みます。
「インポート」を押下すると、ファイル名を入力する画面が表示されるので、CSV形式の加速度センサデータが格納されたファイルを指定して「実行」を押します。
7. 「エクスポート」で、編集した加速度センサデータをCSV形式のファイルに保存します。
「エクスポート」を押下すると、ファイル名を入力する画面が表示されるので、加速度センサデータを保存したいファイルを指定して「実行」を押します。

※インポート・エクスポートの詳細な手順については、「5.18.測位データ編集」を参照してください。

5.25. 動作認識データ編集

i アプリから利用するカメラの動作認識機能の動作認識データを編集します。

動作認識データを編集するには：

1. メニューから「設定」→「ネイティブデータ編集...」→「動作認識データ編集」タブを選択します（図 5-58）。

X方向位置	Y方向位置	品質
100	80	RELIABLE
100	100	LOW_DETAIL

振動量

図 5-58 動作認識データ編集

2. 編集する動作により Roll 動作か Shake 動作のいずれかを選択します。

3. 動作認識データを追加したい場合は、「追加」ボタンを押します。
動作認識データを変更したい場合は、一覧から変更したい動作認識データを選択して、「編集」ボタンを押します（図 5-5 9）。

図 5-5 9 動作認識データ編集画面

4. 各設定項目を設定して「設定」を押します。

各設定項目は表 5-1 5 に示すとおりです。

表 5-1 5 動作認識データの設定項目

設定項目	説明	デフォルト
Roll 動作	X 方向位置：カメラ撮影画像の X 方向位置 Y 方向位置：カメラ撮影画像の Y 方向位置 品質： RELIABLE：信頼できる LOW_DETAIL：変化に乏しく信頼できない TOO_FAST：早すぎて信頼できない SAME_VALUE：前回と同じから選択します。	RELIABLE
Shake 動作	振動量：カメラ撮影画像の振動量 0 ～ 511 の値 0～50：静止状態 50～250：緩やかに振っている 250～400：激しく振っている 400～511：非常に激しく振っている	0

5. 動作認識データを削除したい場合は、一覧から削除したい動作認識データを選択して、「削除」ボタンを押します。
削除してもよいか確認画面が表示されるので「はい」を選択すると削除されます（図 5-57）。

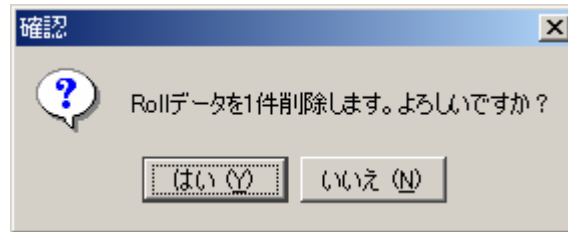


図 5-60 動作認識データ削除確認画面

6. 動作認識データの順序を変更する場合は、順序を変更したい動作認識データを選択して、「上へ」もしくは「下へ」を押します。
7. 「インポート」で、CSV形式の動作認識データを取り込みます。
「インポート」を押下すると、ファイル名を入力する画面が表示されるので、CSV形式の動作認識データが格納されたファイルを指定して「実行」を押します。
8. 「エクスポート」で、編集した動作認識データをCSV形式のファイルに保存します。
「エクスポート」を押下すると、ファイル名を入力する画面が表示されるので、動作認識データを保存したいファイルを指定して「実行」を押します。

※インポート・エクスポートの詳細な手順については、「5.18.測位データ編集」を参照してください。

【Roll 動作認識について】

Roll 動作認識は、カメラから一定のフレームレートで画像をシステム内に取り込み、各フレーム間の画像の差から端末の移動量（ロール量）を取得する機能です。

Roll 動作認識には、以下の2つの移動量取得機能があります。

- 最新の連続2フレーム間での移動量を求める機能 (RollData#getImmediateMotion())
- 基準時点以降の全フレームの移動量を累積し、基準時点から現時点までの移動量を求める機能 (RollData#getAccumulatedMotion())

これらの機能をエミュレートするため、エミュレータでは Roll 動作認識のデータを以下のように使用しております。

- Roll 動作認識のデータは、1件が「1枚の画像」を表します。
例えばフレームレート(*1)が 10fps とした場合、0.1 秒毎に 1 件分のデータを取得するため、1 秒では 10 件のレコードが読み込まれることになります。
(*1)エミュレータでは、「4.5.プロジェクトの設定」で解説するカメラフレームレートの設定に従ってRoll動作認識を行います。
- Roll 動作認識が可能な状態（START 状態）の間、エミュレータは背後で定期的に（フレームレートに応じた間隔で）動作認識データを読み出します。データは先頭から順に読み出され、末尾に達すると再度先頭に戻ってサイクリックに読み出しを続けます。

動作認識データが読み出される度に、そのデータと1つ前のデータの差（x、yそれぞれの差）を算出し、それを該当フレーム間の移動量として保持します。

- **RollReader#getRollData ()**を呼び出すと、基準時点からその時点までの読み込みデータに基づいて算出された全ての移動量が取得されます。具体的には、その際の **RollData** オブジェクトからは以下のような値が得られます。
 - **RollData#getImmediateMotion()**では、**RollData** オブジェクト取得時点でエミュレータが読み込んでいるデータより算出した移動量（x、yの値）のうち、最新の移動量（x、yの値）が得られます。
 - **RollData#getAccumulatedMotion()**では、基準時点から **RollData** オブジェクト取得時点までの間に読み込まれた全データより算出した移動量の累積値（x、yそれぞれの値の累積値）が得られます。

DoJa仕様と同様、エミュレータにおいても基準時点は **RollReader#getRollData()**や **RollReader#recenter()**を呼び出す度に、その時点で読み出されている最新のデータの位置に再設定されます。

5.26. ビルドの設定

5.26.1. スクリプトの実行

i アプリをビルドする際に、リバースエンジニアリングなどで最適化をする事によって、i アプリの Jar ファイルのサイズをコンパクトにしたい場合があります。

エミュレータでは、これを支援する機能として、ビルドの実行前か実行後に i アプリ (Jar) が作成されるパスでスクリプトコマンドを実行できるようにしています (図 5-6 1)。

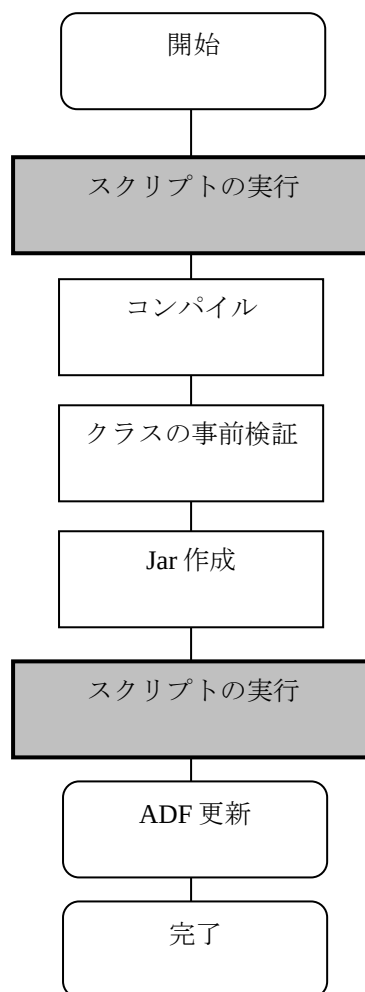


図 5-6 1 ビルド時のスクリプトコマンド実行の流れ

ビルドの設定でスクリプトを実行できるようにするには：

1. メニューから「設定」→「エミュレータ環境設定...」→「ビルドの設定」タブを選択します（図 5-6 2）。

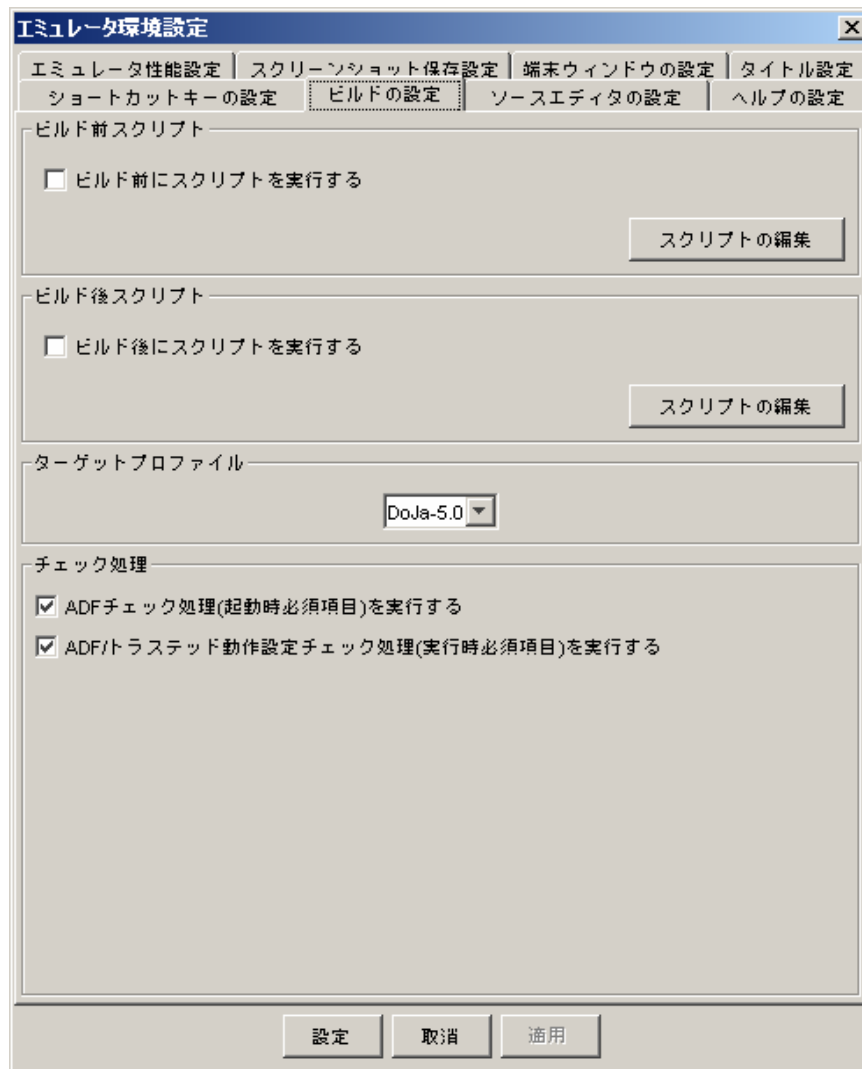


図 5-6 2 ビルドの設定 - スクリプトの実行

2. ビルドの実行前にスクリプトを実行したい場合は、「ビルド前にスクリプトを実行する」にチェックを付けます。
ビルドの実行後にスクリプトを実行したい場合は、「ビルド後にスクリプトを実行する」にチェックを付けます。
3. 「スクリプトの編集」を押すとエディタでスクリプトファイルが開かれるので、ここに DOS のバッチファイル形式で起動したいコマンドや必要な環境変数を記述します。
4. 「設定」もしくは「適用」を押します。

スクリプトの内容には、次の環境変数が使えます：

環境変数	内容
PROJECT_NAME	ビルドするプロジェクト名
JAR_PATH	jar コマンド (i アプリの Jar ファイルを操作する際に使用する)
APP_CLASS	start() メソッドが存在するクラス名

スクリプトの終了は、次のように指定してください。

■ 正常終了の場合

`exit 0`

■ 異常終了の場合 (メッセージを表示してビルドを中断します。)

`exit 1` (0 以外の値)

スクリプトは、以下のファイルで作成されます：

■ ビルド前スクリプト：

`C:\iDKDoJa5.0\lib\optimize_script1.bat`

■ ビルド後スクリプト：

`C:\iDKDoJa5.0\lib\optimize_script2.bat`

5.26.2. ターゲットプロファイルの設定

i アプリをビルドする時の DoJa のプロファイルを設定します。
デフォルトは、最新の「DoJa-5.0」になっています。

ビルド時の DoJa のプロファイルを設定するには：

1. メニューから「設定」→「エミュレータ環境設定...」→[ビルドの設定] タブを選択します（図 5-6 3）。

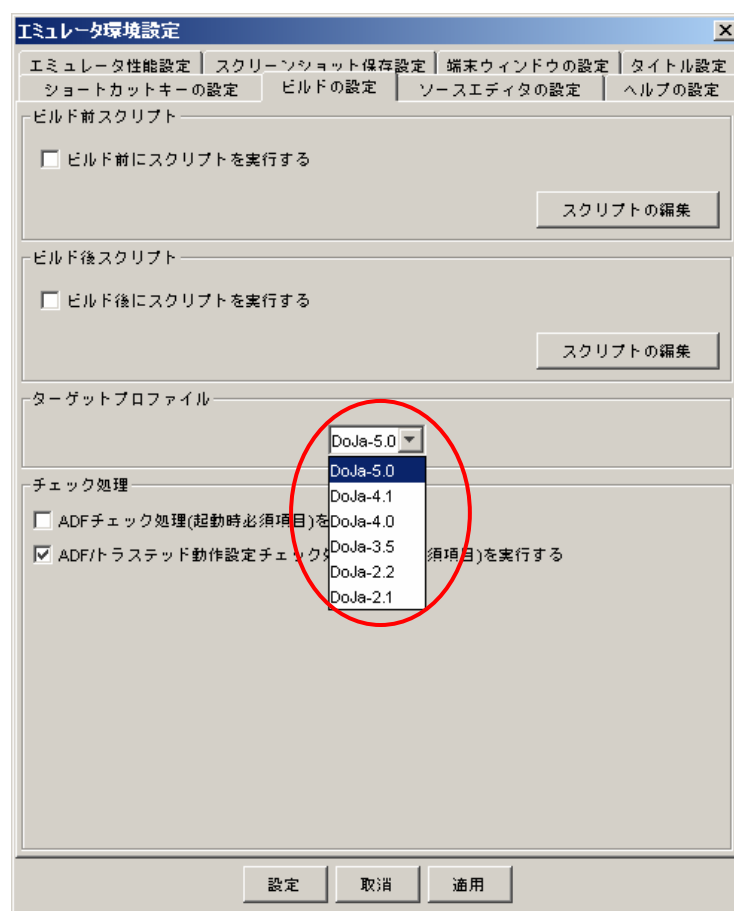


図 5-6 3 ビルドの設定 - プロファイルの変更

2. ターゲットにしたい DoJa のプロファイルを選択します。
3. 「設定」もしくは「適用」を押します。
これにより、ビルド時にターゲットプロファイルに含まれないAPIを用いていた場合は、コンパイルエラーになります。

【注意】

ここでプロファイルを切り替えても i アプリの実行ではチェックされません。
この設定は、i アプリをビルドする際に、選択したターゲットプロファイルを使います。

5.27. ADF・トラステッド動作設定チェック処理の設定

i アプリのビルド時に ADF のキーおよびトラステッド動作設定の設定が不足していないかをチェックすることができます。

インストール直後は、このチェックを実行するようになっています。

もし、ビルドにかかる時間が気になるようでしたらチェックしないように変更してください。

ADF・トラステッド動作設定チェックの設定を変更するには：

1. メニューから「ツール」→「エミュレータ環境設定...」→「ビルドの設定」タブを選択します（図 5-64）。

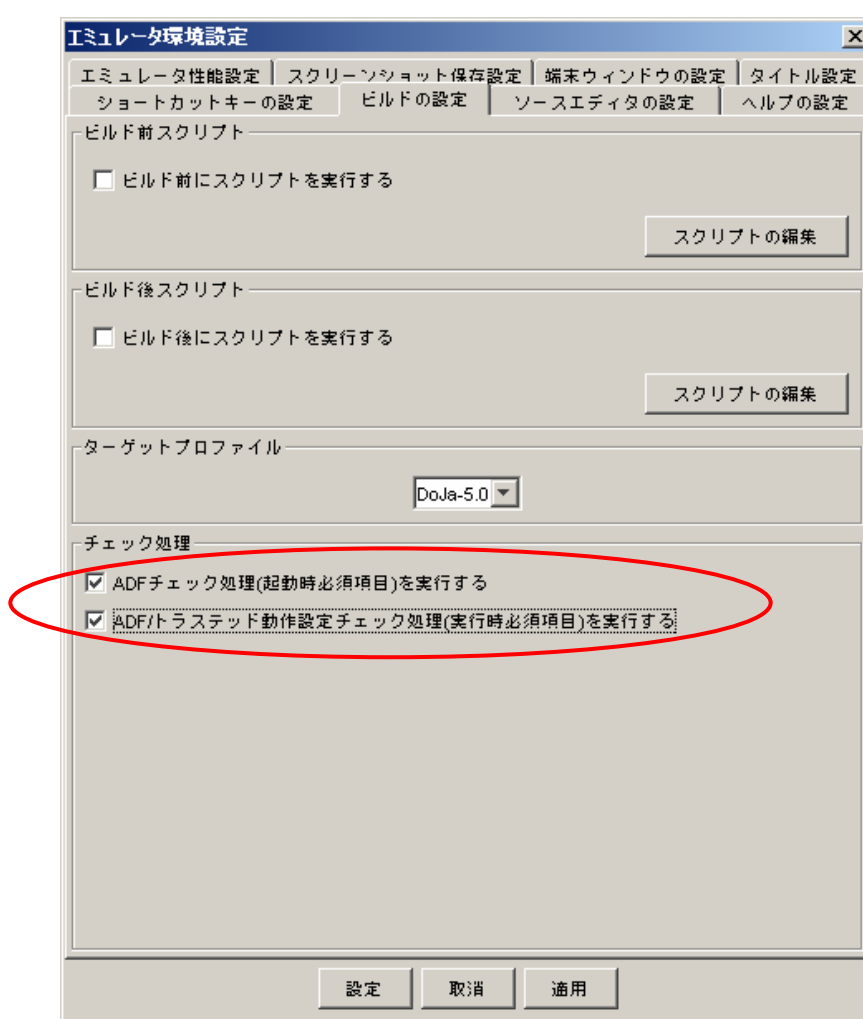


図 5-64 ADF・トラステッド動作設定チェック処理

2. ビルド時に i アプリの起動許可 をチェックしたい場合は、「ADFチェック処理（起動時必須項目）を実行する」にチェックを付けます。
ビルド時に A P I の実行許可 をチェックしたい場合は、「ADF／トラステッド動作設定チェック処理（実行時必須項目）を実行する」にチェックを付けます。

3. 「設定」もしくは「適用」を押します。
 これにより、ビルド時に必要なAPIを構文解析して警告を表示します（図 5-6 5）。

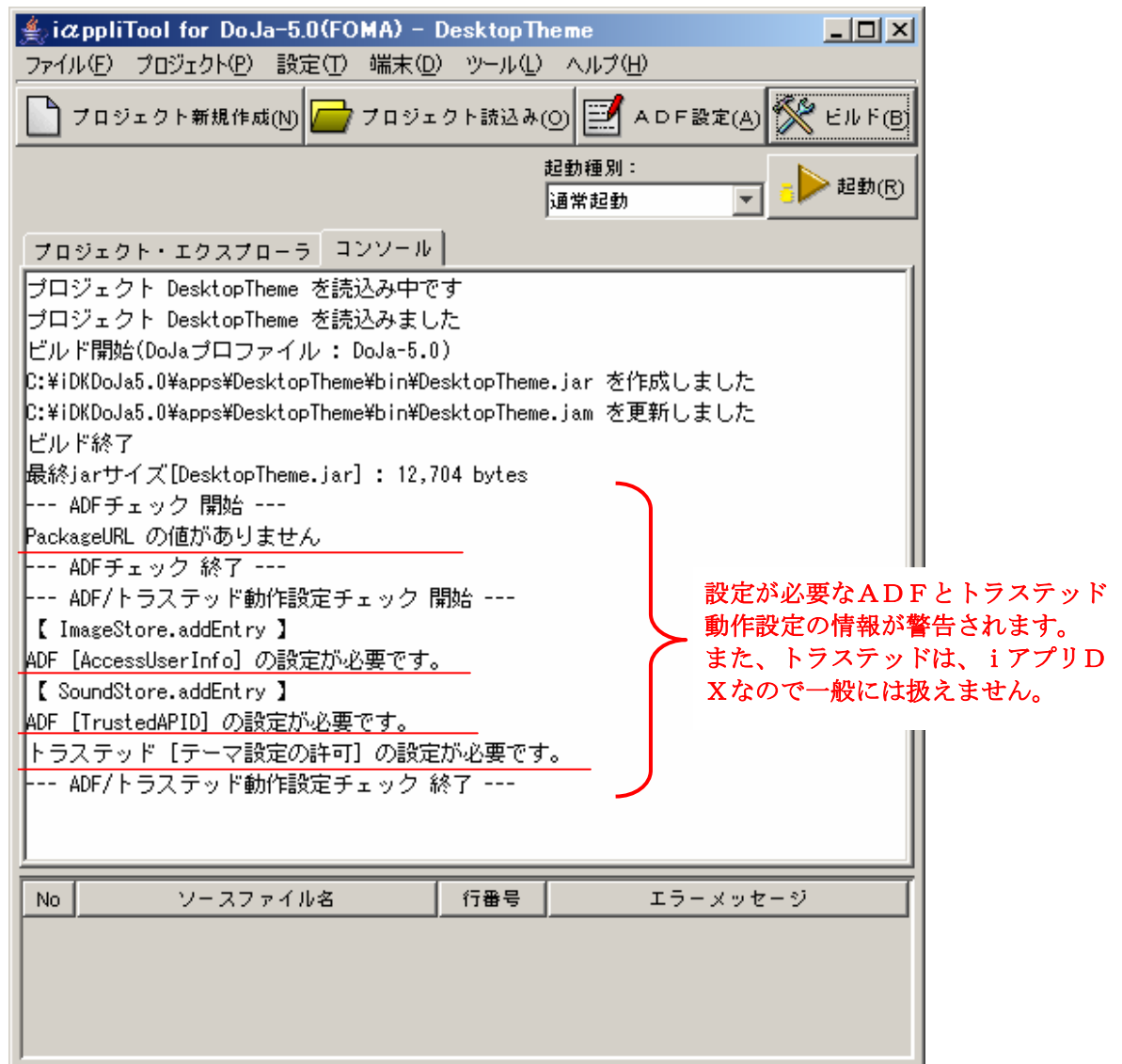


図 5-6 5 ADF・トラステッド動作設定のチェック結果

5.28. ショートカットキーの設定

エミュレータで扱うショートカットキーを設定します。

エミュレータでは、インストールされた状態で、あらかじめ以下のショートカットキーが設定されています：

■エミュレータ (iα ppliTool)

コンソールクリア：	C
プロジェクト新規作成：	N
プロジェクト読込み：	O
ソースファイルを開く：	F
A D F 設定：	A
ビルド：	B
起動：	R

■エミュレーション端末

数字ボタン：	0 ～ 9
# ボタン：	#
* ボタン：	*
決定ボタン：	Enter (↵)
方向ボタン：	Up (↑) , Down (↓) , Left (←) , Right (→)
待受ボタン：	D
ソフトキー 1：	A (テキスト編集中は無効)
ソフトキー 2：	S (テキスト編集中は無効)

※ エミュレーション端末のショートカットキーを変更することはできません。

エミュレータのショートカットキーを設定するには：

1. メニューから「設定」→「エミュレータ環境設定...」→「ショートカットキーの設定」タブを選択します（図 5-6 6）。

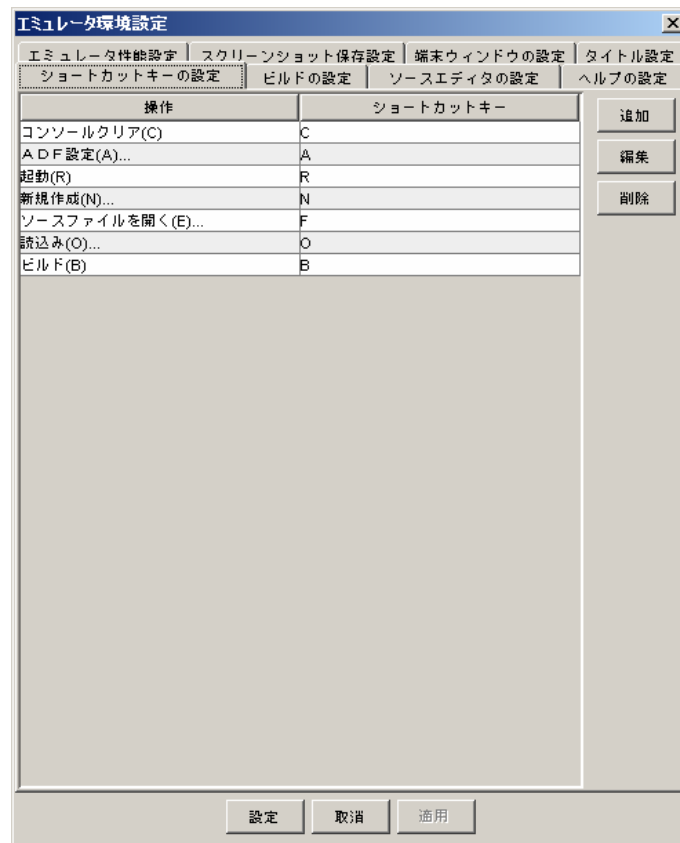


図 5-6 6 ショートカットキーの設定

2. ショートカットキーを追加する場合は、「追加」ボタンを押します（図 5-6 7）。ショートカットキーを変更する場合は、変更したいショートカットキーを選択して「編集」ボタンを押します。
※ 同じ操作に複数のショートカットキーを割り当てることはできません。

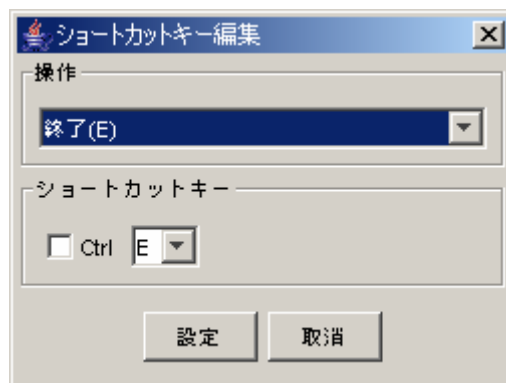


図 5-6 7 ショートカットキーの編集

3. 各設定項目を入力して、「設定」もしくは「適用」を押します。

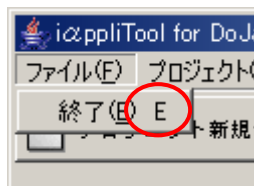
各設定項目は表 5-16 に示すとおりです。

表 5-16 ショートカットキーの設定項目

設定項目	説明	デフォルト
操作	ショートカットキーを設定したい操作を選択します。 既に設定済みの同じ操作を指定することはできません。	C: コンソールクリア N: プロジェクト新規作成 O: プロジェクト読み込み F: ソースファイルを開く
ショートカットキー	設定するショートカットキーを選択します。Ctrl にチェックを付けるとショートカットは、Ctrl + キー になります。	A: A D F 設定 B: ビルド R: 起動

4. ショートカットキーの一覧に設定したショートカットが追加されます。

ショートカットキーを設定するとiα ppliToolのメニューに設定したショートカットキーが表示されます。(図 5-68)



エミュレータ上で「e」または「E」を押すとエミュレータが終了します。

図 5-68 終了メニューに「E」のショートカットキーが設定された状態

5.29. ヘルプの設定

エミュレータのヘルプメニューからAPIリファレンス、操作ガイド（本書）およびリリースノートを参照する設定をします。

ヘルプの設定をするには：

1. メニューから「設定」→「エミュレータ環境設定...」→「ヘルプの設定」を選択します（図 5-69）。

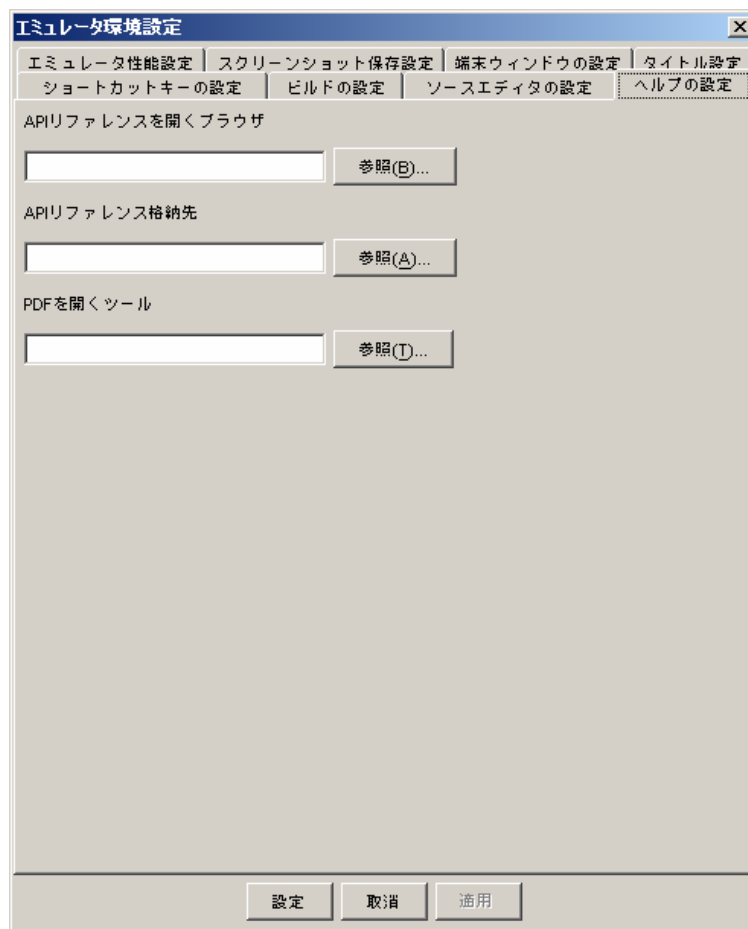


図 5-69 ヘルプの設定画面

2. 各設定項目を入力して、「設定」もしくは「適用」を押します。

各設定項目は表 5-17 に示すとおりです。

表 5-17 ヘルプの設定項目

設定項目	説明	デフォルト
API リファレンスを開く ブラウザ	API リファレンスを参照するためのブラウザを設定します。	インストールされた Internet Explorer もしくは、 Netscape
API リファレンス格納先 (※)	API リファレンスが格納されているディレクトリを設定します。 ■API リファレンス： API リファレンス格納先¥index.html を開きます。	インストールディレクトリ¥doc
PDF を開くツール	PDF ファイルを参照するための i アプリを設定します。 ■操作ガイド： インストールディレクトリ¥～DevGuide～.pdf ■リリースノート： インストールディレクトリ¥～ReleaseNote～.pdf を開きます。	インストールされた Acrobat Reader

※ 別途 <http://www.nttdocomo.co.jp/service/imode/make/index.html> から入手してください。

5.30. エミュレータ性能設定

エミュレータの性能を変えることで、エミュレータを携帯端末の性能に近づけることができます。

エミュレータの性能を設定するには：

1. メニューから「設定」→「エミュレータ環境設定...」→「エミュレータ性能設定」を選択します（図 5-7 0）。

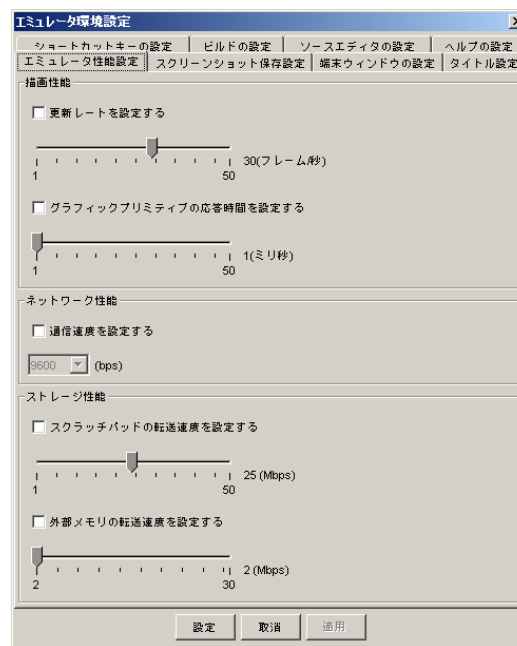


図 5-7 0 エミュレータ性能設定画面

2. 各設定項目を設定して、「設定」もしくは「適用」を押します。

各設定項目は、設定したい項目にチェックを付けるとスライドバーで変更できるようになっています。

チェックを外すと性能の設定は無効になります。

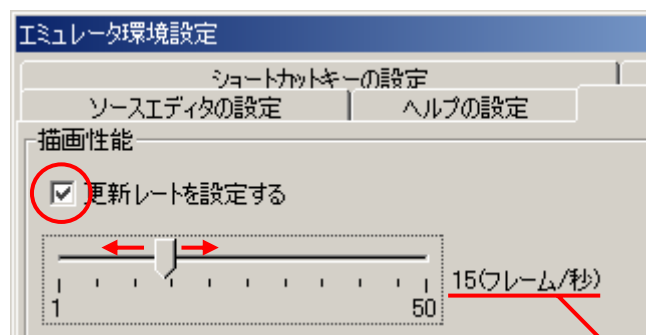


図 5-7 1 性能値の変更

現在の値

各設定項目は表 5-18 に示すとおりです。

表 5-18 エミュレータ性能の設定項目

性能項目	設定項目	説明	デフォルト
描画性能	更新レートを設定する	Graphics のフレームレートを 1 ～ 50 fps の範囲で変更します。 未設定の場合、77 フレーム/秒。	30 フレーム/秒
	グラフィックプリミティブの応答時間を設定する	Graphics の各プリミティブ描画の待ち時間を 1 ～ 50 ms の範囲で設定します。これにより 描画速度を遅らせてデバッグ することが可能です。 未設定の場合、0 ミリ秒。	1 ミリ秒
ネットワーク性能	通信速度を設定する	HTTP 通信の待ち時間を 1 ～ 50 bps の範囲で設定します。 未設定の場合、デバイス依存。	9600 bps
ストレージ性能	スクラッチパッドの転送速度を設定する	スクラッチパッドの読み書きにかかる待ち時間を 1 ～ 50 Mbps の範囲で設定します。 未設定の場合、PC 性能に依存。	25 Mbps
	外部メモリの転送速度を設定する	外部メモリの読み書きにかかる待ち時間を 2 ～ 30 Mbps の範囲で設定します。(最小は 2 Mbps) 未設定の場合、PC 性能に依存。	2 Mbps

5.31. エミュレーション端末ウィンドウの設定

エミュレーション端末のウィンドウは、実際の i モード携帯電話に合ったサイズに変更することができます。^[7]

エミュレーション端末のサイズを設定するには：

1. メニューから「設定」→「エミュレータ環境設定...」→「端末ウィンドウの設定」を選択します（図 5-7 2）。

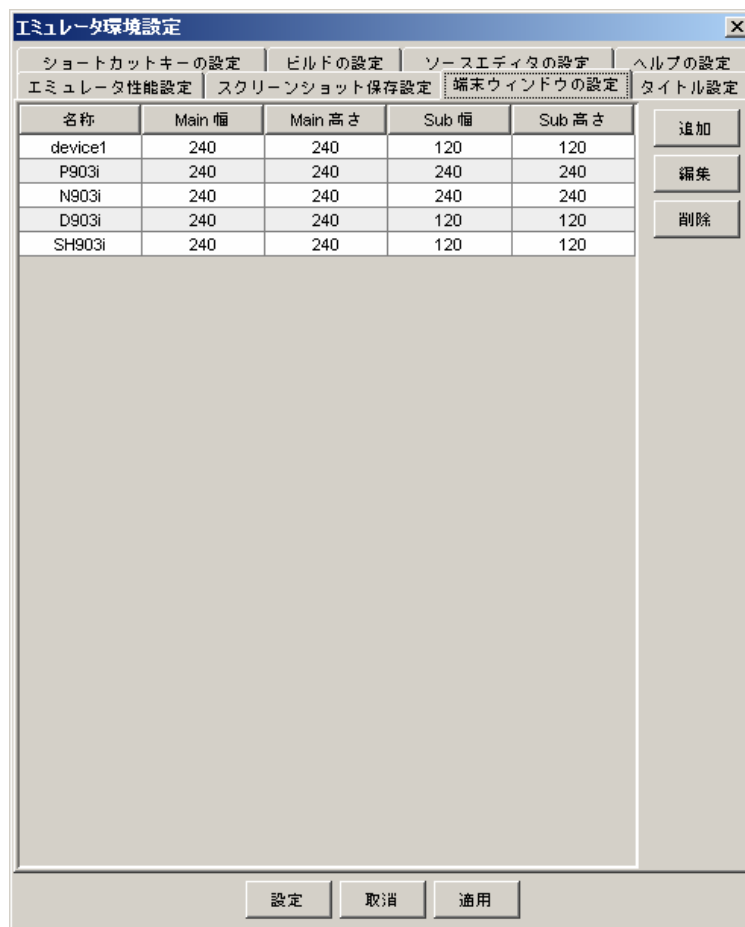


図 5-7 2 エミュレーション端末ウィンドウ一覧画面

2. 現在設定されているエミュレーション端末が一覧表示されます。
 初期状態では、**device1** のみで、この **device1** については編集できません。
 エミュレーション端末を追加する場合は、「追加」ボタンを押します（図 5-7 3）。
 エミュレーション端末を変更する場合は、変更したいエミュレーション端末を選択して「編集」ボタンを押します。

^[7] Display#getWidth()及び Display#getHeight() は、ここで設定した値と同じになります。

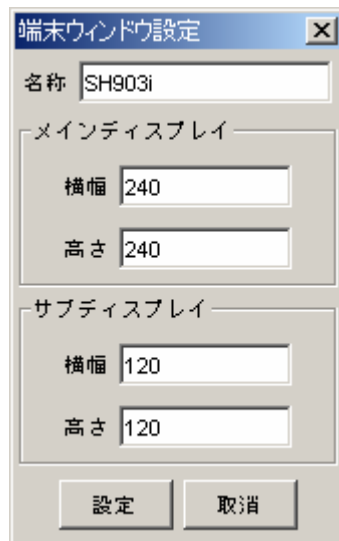


図 5-7 3 エミュレーション端末ウィンドウ編集画面

3. 各設定項目を設定して、「設定」もしくは「適用」を押します。

各設定項目は表 5-1 9 に示すとおりです。

表 5-1 9 エミュレーション端末ウィンドウの設定項目

設定項目	説明	デフォルト
名称	エミュレーション端末の名称を指定します。 ここで指定した名称が端末メニューに表示されます。既に登録されたものと同じ名前を付けることはできません。	なし
メインディスプレイ 横幅	メインディスプレイの横幅を 240 ～ 480 の範囲で指定します。	なし
メインディスプレイ 高さ	メインディスプレイの高さを 240 ～ 640 の範囲で指定します。	なし
サブディスプレイ 横幅	サブディスプレイ(背面液晶)の横幅を 120 ～ 240 の範囲で指定します。	なし
サブディスプレイ 高さ	サブディスプレイ(背面液晶)の高さを 120 ～ 320 の範囲で指定します。	なし

5.32. タイトル設定

エミュレータのタイトルを変更することができます。

タイトルは、「製品名 - プロジェクト名」、「プロジェクト名 - 製品名」のいずれかを選べます。「プロジェクト名 - 製品名」に設定しておくで、最小化のときに開かれているプロジェクトが分かりやすくなります。

エミュレータのタイトルを変更するには：

1. メニューから「設定」→「エミュレータ環境設定...」→「タイトル設定」を選択します（図 5-74）。



図 5-74 タイトル設定

2. 「製品名 - プロジェクト名」、「プロジェクト名 - 製品名」のいずれかを選択して、「設定」を押します。

5.33. スクリーンショット保存の設定

エミュレーション端末のスクリーンショットの保存方法を変更することができます。
スクリーンショットの詳しい説明については、「6.6. スクリーンショット」を参照してください。

エミュレーション端末のスクリーンショットの保存方法を変更するには：

1. メニューから「設定」→「エミュレータ環境設定...」→「スクリーンショット保存設定」を選択します（図 5-7 5）。

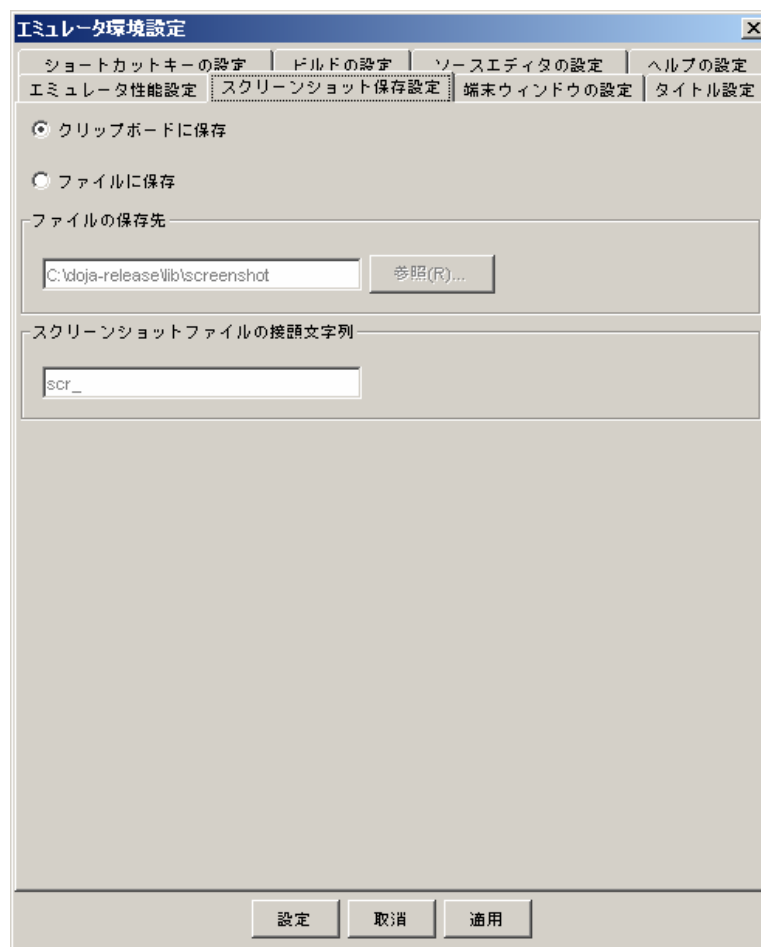


図 5-7 5 スクリーンショット保存設定

2. クリップボードに保存する場合は、「クリップボードに保存」を選択します。
ファイルに保存する場合は、「ファイルに保存」を選択します。
デフォルトは、クリップボードになっています。
3. ファイルに保存する場合、保存先のパスとファイル名の接頭語を指定します。
ファイルは、「接頭語 XXXXXXXX.bmp」(XXXXXXX:連番)で保存されます。
4. 「設定」もしくは「適用」を押します。

5.34. コンパイラ(javac)の選択

メニューから「設定」→「sun.tools.javac.Mainを使用する」を選択することにより、コンパイラ (javac) を選択することができます (図 5-76)。ここで選択されたコンパイラ (javac) は、プロジェクトのビルド時 (「4.3. プロジェクトのビルドと通常起動」を参照) に使用されるコンパイラ (javac) です。

デフォルトでは「com.sun.tools.javac.Main」を使用しますが、「sun.tools.javac.Mainを使用する」チェックボックスをチェックすることにより「sun.tools.javac.Main」を使用します。

また、選択されたコンパイラ (javac) の情報は保存されるので、次回「iαppliTool」起動時に自動的に設定されます。

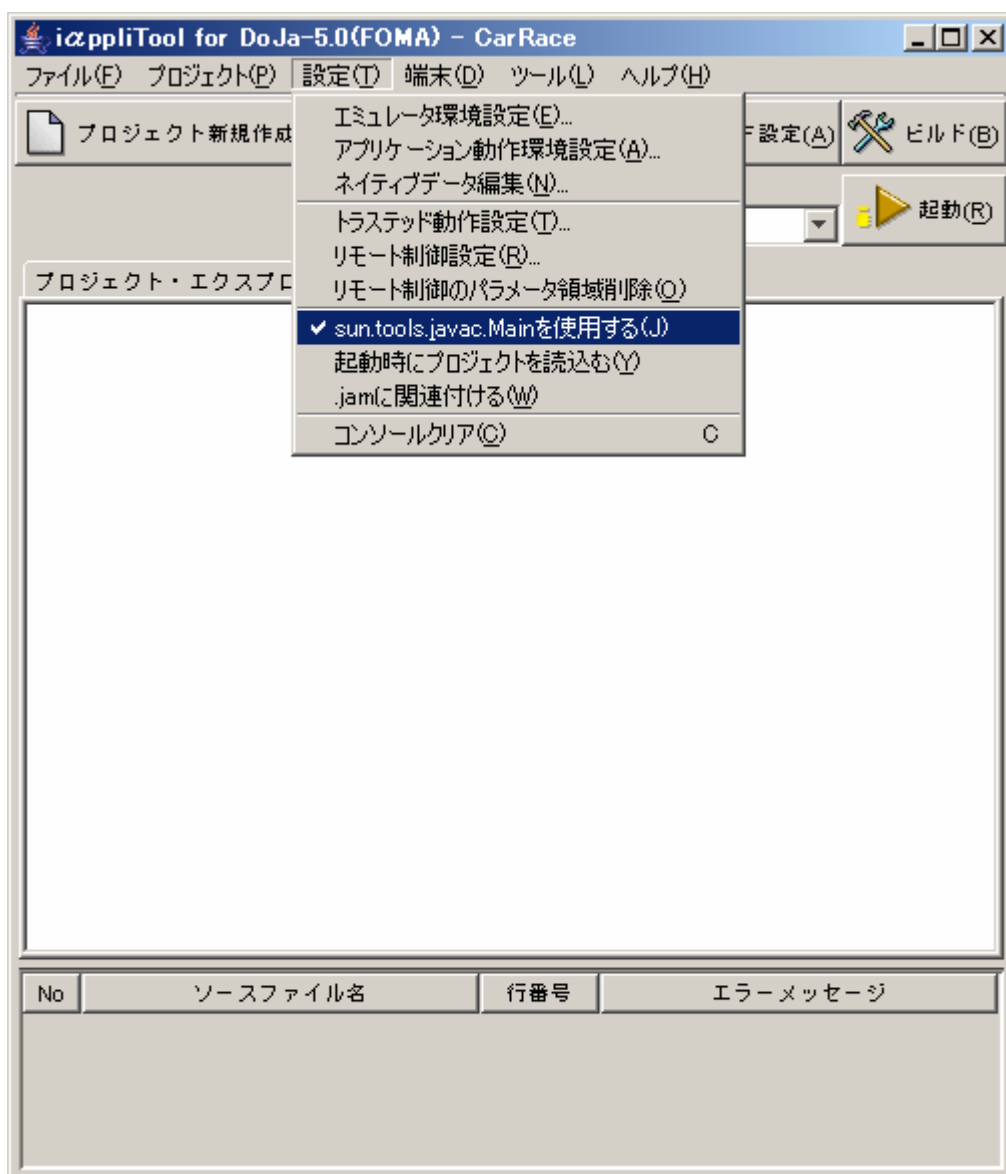


図 5-76 コンパイラ (javac) の選択

5.35. 起動時にプロジェクトを読み込む

エミュレータが起動した時に、前回の終了時に開いていたプロジェクトを開く事ができます。これによりプロジェクトを開きなおす手間が省けます。

プロジェクトが開いた状態でエミュレータを起動するには、メニューから「設定」→「起動時にプロジェクトを読み込む」にチェックを付けます（図 5-7 7）。

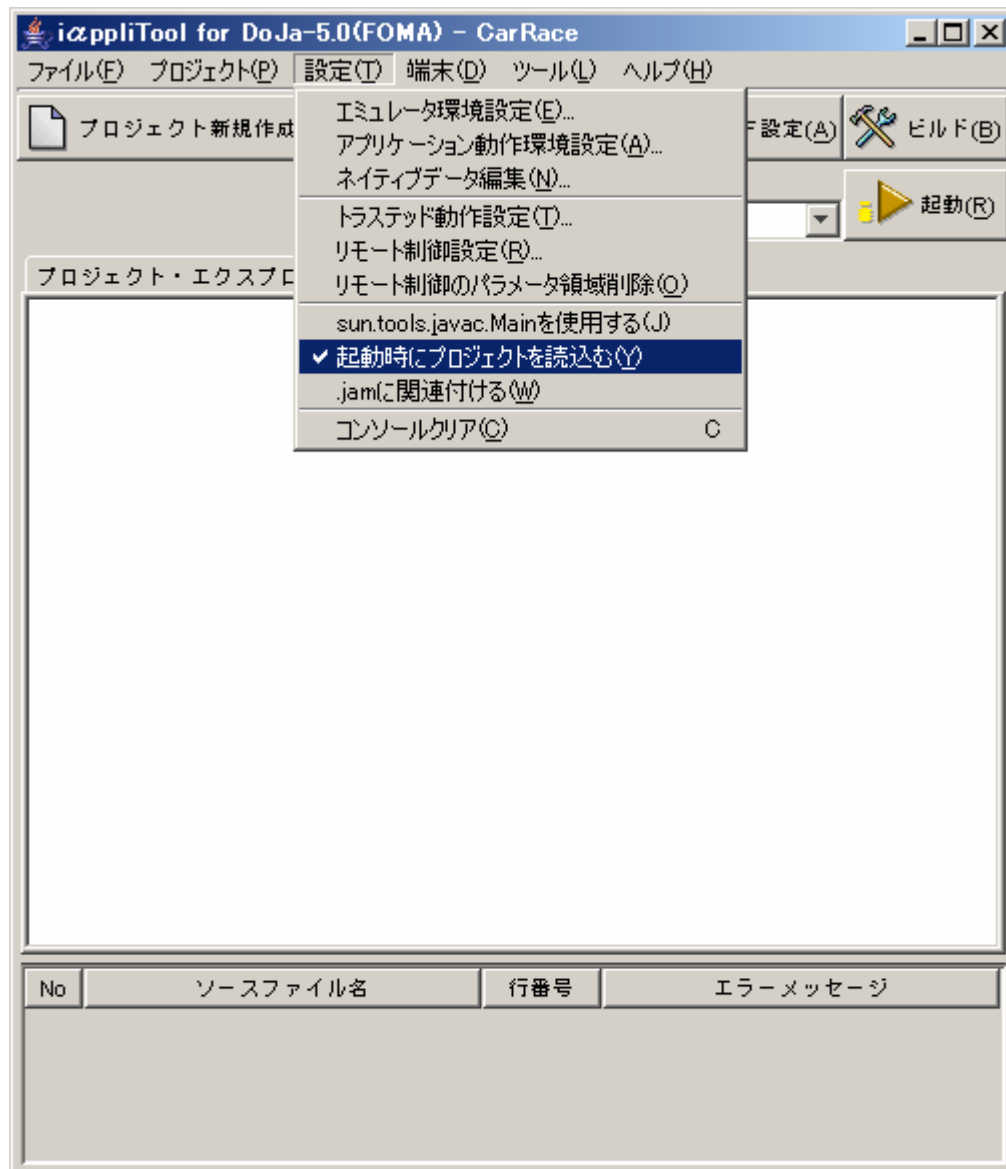


図 5-7 7 起動時にプロジェクトを読み込む

5.36. jam に関連付ける

ADF（.jam ファイル）をエミュレータに関連付けします。
 これにより、ADF ファイルをクリックするだけで、エミュレータが起動し、i アプリが実行されます。
 この機能は、インストール時も設定するかどうかを問い合わせています。

ADF（.jam ファイル）をエミュレータに関連付けるには、メニューから「設定」→「jam に関連付ける」にチェックを付けます（図 5-78）。

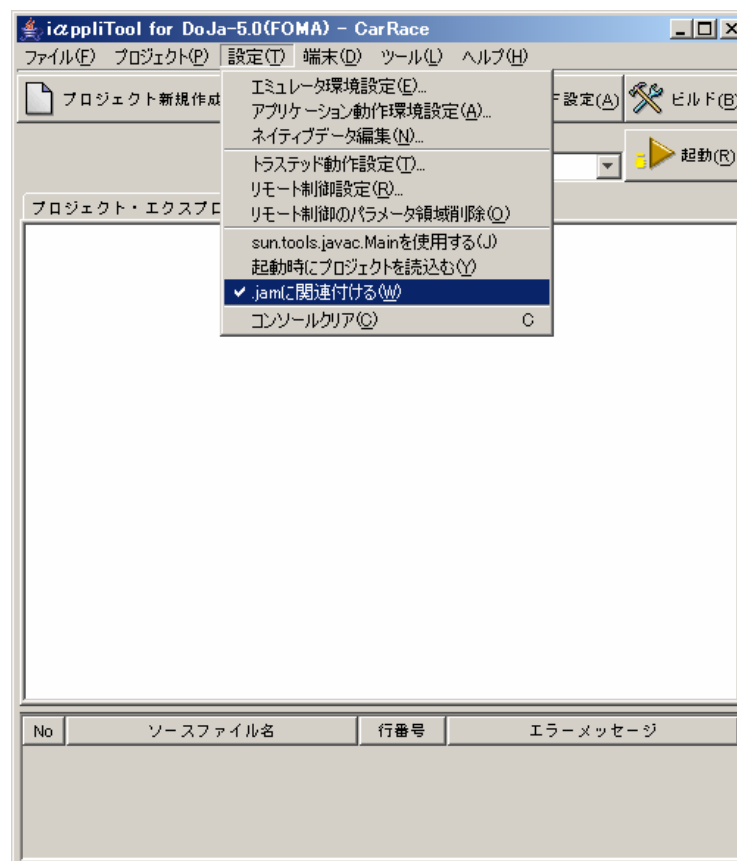


図 5-78 .jam に関連付ける

5.37. コンソールクリア

起動中の i アプリからの標準出力・エラー出力は、コンソールに表示されます^[8]。
 メニューから「設定」→「コンソールのクリア」を選択すると、コンソールに表示されているメッセージが消去されます。

^[8] ただし、絵文字は「・」、JIS X 0208 で定義されている漢字以外の漢字は「?」で表示されます。

5.38. エミュレーション端末の選択

i アプリが動作するエミュレーション端末を変更することができます。

i アプリが動作するエミュレーション端末を変更するには：

1. メニューから「端末」を選択し、その中からエミュレートする端末をチェックします（図 5-79）。

このメニューには、ビルドした i アプリに対して使用できる端末が表示されます。なお、ここで選択した端末は、レジストリに保存されるので、次回「iα ppliTool」を起動したとき最初に選択されている端末として使用されます。

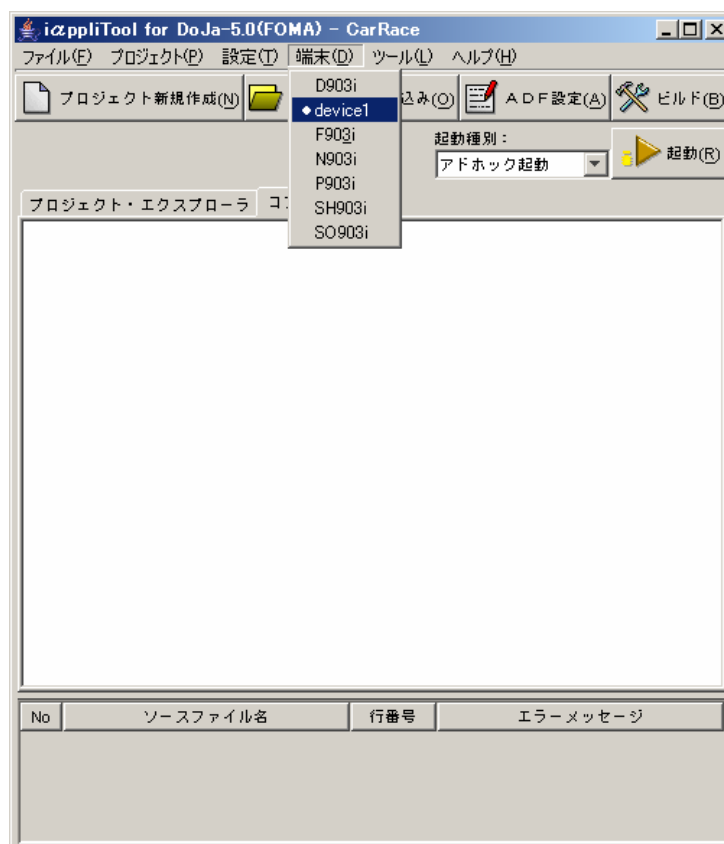


図 5-79 端末の選択

💡 ワンポイント

エミュレーション端末は、携帯端末の機種に合わせて異なるウィンドウサイズのものを複数作成することが可能です。
エミュレーション端末の設定については、「5.31. エミュレーション端末ウィンドウの設定」を参照してください。

5.39. エミュレーション端末の操作

次にエミュレーション端末の基本操作を説明します（図 5-80）。
ここで説明している基本操作はどのエミュレーション端末でも同様です。

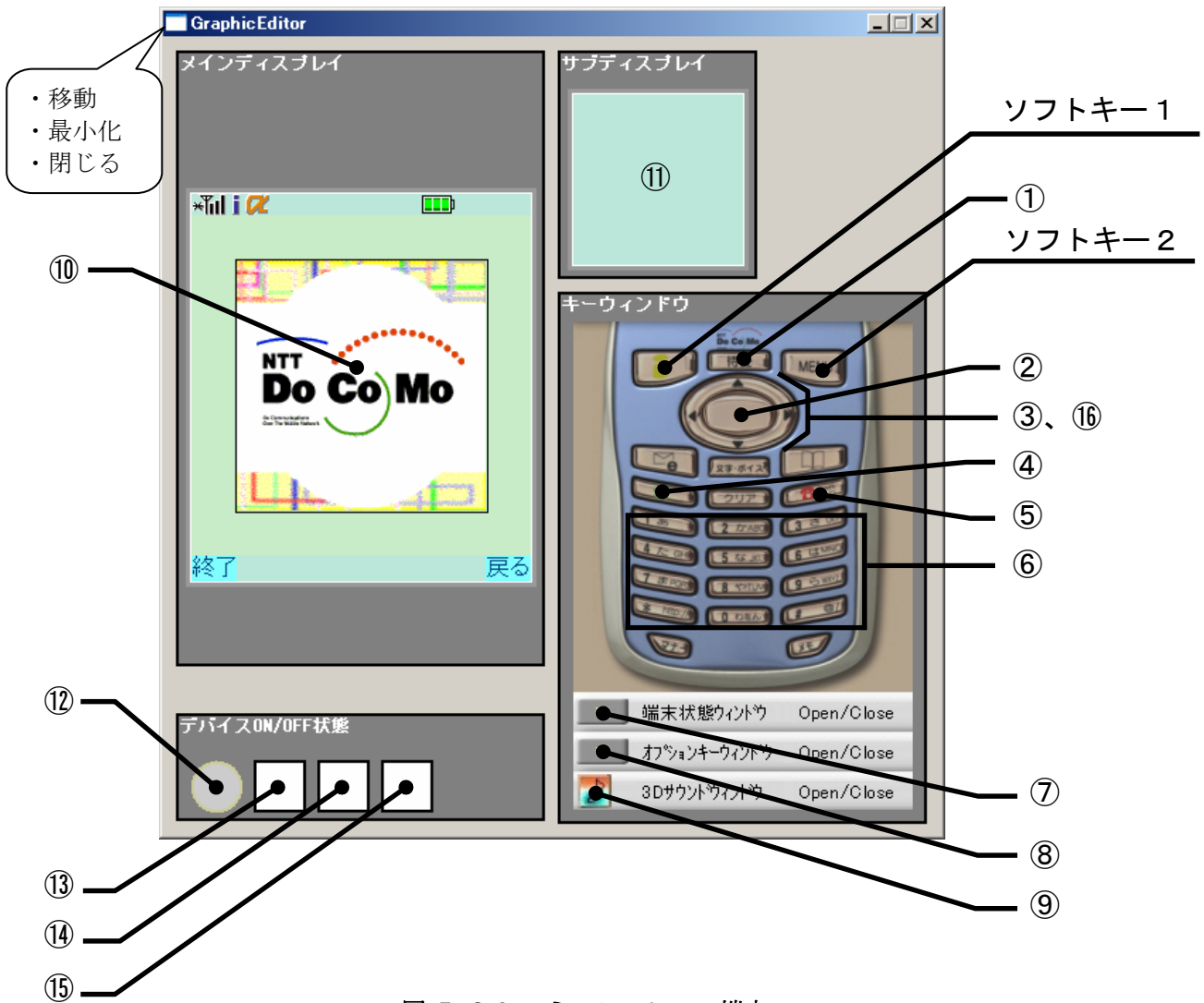


図 5-80 エミュレーション端末

①. 待受ボタン

- 待ち受けアプリケーションが、「休眠状態」あるいは「非活性化状態」のときに押すことにより「活性化状態」になります（「4.8.1.待ち受けアプリケーション」参照）。このボタンは、キーボードの「D」キーに対応しています。

②. 決定ボタン

- このボタンは、キーボードの「Enter（リターン）」キーに対応しています。

③. 方向ボタン（上下左右ボタン）

- このボタンは、キーボードの「上下左右（↑↓←→）」キーにそれぞれ対応しています。

④. 発信ボタン

- 通常の実用アプリケーションおよび待ち受けアプリケーション動作中に押すことにより、通常の通話を開始した状態になり、i アプリがサスペンド(中断)します。

⑤. 終話ボタン

- ①の発信ボタンを押した状態（通話中）のときに押すことにより、通話が終了し、サスペンド(中断)していたi アプリがレジューム(復帰)します。このときレジューム(復帰)した事がi アプリに通知されます。実際のi モード携帯電話では、サスペンドーレジュームは通話中だけでなく、メニューの切り替えやダイアログの表示などでも発生します。
- 上記の状態以外のときに押すことにより、通常の実用アプリケーションおよび待ち受けアプリケーションを強制終了させます。

⑥. 数字ボタン（0～9 ボタン）

- このボタンは、キーボードの「0～9」キーにそれぞれ対応しています。

⑦. 端末状態ウィンドウ表示

ここをクリックすると、端末の状態を変更する画面を表示します。
もう一度クリックすると画面は閉じられます。
端末状態ウィンドウについては、「5.3 9.1. 端末状態ウィンドウの操作」で詳しく説明します。

⑧. オプションキーウィンドウ表示

エミュレーション端末にないキーイベントを発生させる画面を表示します。
もう一度クリックすると画面は閉じられます。
オプションキーウィンドウについては、「5.3 9.2. オプションキーウィンドウの操作」で詳しく説明します。

⑨. 3Dサウンド表示

3Dサウンドを演奏している時に視覚的に演奏している位置を確認できる画面を表示します。

3Dサウンド表示については、「5.3 9.3. 3Dサウンド表示」で詳しく説明します。

⑩. LCD

- i アプリ起動中に LCD 上でマウスを右クリックすることで、エミュレーション端末の状態を変更する事ができます。また、エミュレーション端末に無いイベントを発生する事ができます。
ただし、API から状態を変更した場合は、チェックのついた属性とは異なる場合があります。
状態とイベントには、それぞれ以下に示す属性があります:

■端末状態

- 「折りたたみ状態」
 - 開く
 - 閉じる
- 「画面状態」
 - 可視
 - 不可視
- 「メール状態」
 - メールなし
 - 未読メールあり
 - センターに未読メールあり
- 「メッセージ状態」
 - メッセージなし
 - 未読メッセージあり
 - センターに未読メッセージあり
- 「バッテリー状態」
 - バッテリー充電中
 - バッテリーMAX
 - バッテリーその他
- 「アンテナ状態」
 - アンテナ圏外
 - アンテナ圏内
- 「マナーモード状態」 (i アプリ起動中は設定不可)
 - マナーモードOFF
 - マナーモードON

■ 端末ウィンドウ以外で発生できるイベント

- 「MP4 再生終了」
- 「MP4 ストリーミング終了」
- 「メッセージフォルダイベント」
 - 「受信フォルダ」
 - 「送信フォルダ」
 - 「未送信フォルダ」
- 「パラメータ受信」
- 「FeliCa アドホック接続待ち」

このメニューを選択すると、コンソール上に「アドホック連携待機開始しました」と表示され、アドホック起動と同じく、現在起動している i アプリでアドホック接続を待ち受けます。「アドホック連携待機終了しました」と表示されるまでの間（1 分間）相手からの接続 (AdhocDataTransfer#setup) を待ち受けます。
- 「FeliCa Push イベント通知」

このメニューを選択すると、外部 R/W からの Push 通知をエミュレーションします。Felica#setFelicaPushListener にて FelicaPushListener が登録されている場合は、リスナに Push イベントが通知されます。
- 「GPS 測位開始(native)」

このメニューを選択すると、「GPS 測位中」の画面に切り替わり、実行中のアプリケーションはサスペンドされます。

これにより、ネイティブで実行される GPS の定期測位との競合を試験する事ができます。
- 「メインスクリーンの変化」

このメニューを選択すると、メインスクリーンの向きの変化を加速度センサ機能の AccelerationEventListener#actionPerformed() に通知します (向きの変化を 0、90、180、270 度から選択します)。
- 上記の状態を i アプリから取得するには、PhoneSystem#getAttribute(int attr)メソッドを使用します（詳細は、API ドキュメント参照）。なお、「メール状態」、「メッセージ状態」、「バッテリー状態」、「アンテナ状態」、および「マナーモード状態」をこのメソッドで取得する場合、ADF の設定項目「GetSysInfo」を設定しておく必要があります（「4.7. アプリケーションディスクリプタファイル (ADF) の編集」を参照）。

⑪. サブディスプレイ

i アプリオプション API SubDisplay を使った場合のサブディスプレイ(背面液晶)の表示します。

サブディスプレイがあるかどうかは携帯端末の種類に依存します。

⑫. ピクチャライト

i アプリオプション API PictureLight を使った場合に発光します。

⑬. バイブレータ状態

バイブレータを ON にすると表示されます。



⑭. バックライト状態

バックライトをONにすると表示されます。



⑮. FeliCa 搬送波

FeliCa のデータ通信をすると表示されます。



⑯. ポインティングデバイスの実行

i アプリオプション A P I PointingDevice を使った場合に有効になります。

■ ジョイスティックモード：

マウスの右ボタンをドラッグしながら決定ボタンの上を移動させると移動量が取得できます。

■ マウスモード：

マウスの右ボタンをドラッグしながら決定ボタンの上を移動させるとLCDにポインタが現われて移動量が取得できます。



*** PointingDeviceクラス ***

ポインタ



マウスの右ボタンをドラッグしながら
決定キーの上を移動する



■ ソフトキー1（左側）、ソフトキー2（右側）

キーボードの「A」「S」キーにそれぞれ対応しています（ただし、テキスト編集中は無効です）。

5.39.1. 端末状態ウィンドウの操作

端末状態ウィンドウは、エミュレーション端末の端末状態ウィンドウ表示を押すと表示されます（5.39. エミュレーション端末の操作 - ⑦ 参照）。

The screenshot shows a window titled '端末状態ウィンドウ' (Terminal Status Window) with several sections for configuring the terminal's status:

- 折りたたみ状態** (Folded state): Radio buttons for ☒ 開く (Open) and ☐ 閉じる (Close).
- 画面状態** (Screen state): Radio buttons for ☒ 可視 (Visible) and ☐ 不可視 (Invisible).
- マナーモード状態** (Manner mode state): Radio buttons for ☐ マナーモードON (Manner mode ON) and ☒ マナーモードOFF (Manner mode OFF).
- メール状態** (Email status): A dropdown menu currently showing 'メールなし' (No email).
- メッセージ状態** (Message status): A dropdown menu currently showing 'メッセージなし' (No message).
- バッテリー状態** (Battery status): A dropdown menu currently showing 'バッテリーMAX' (Battery MAX).
- アンテナ状態** (Antenna status): A dropdown menu currently showing 'アンテナ圏内' (Antenna in range).

図 5-8 1 端末状態ウィンドウ

端末状態ウィンドウでは、エミュレーション端末の状態を変更する事ができます。この状態は、エミュレーション端末のLCD上でも同様に変更する事ができます。状態を変更するには、それぞれの状態を選択します。また、iアプリ実行前に状態を設定した場合、その状態がエミュレーション端末に反映されます。

- 「折りたたみ状態」
 - 開く
 - 閉じる
- 「画面状態」
 - 可視
 - 不可視
- 「メール状態」
 - メールなし
 - 未読メールあり
 - センターに未読メールあり
- 「メッセージ状態」
 - メッセージなし
 - 未読メッセージあり
 - センターに未読メッセージあり
- 「バッテリー状態」
 - バッテリー充電中
 - バッテリーMAX
 - バッテリーその他

- 「アンテナ状態」
 - アンテナ圏外
 - アンテナ圏内
- 「マナーモード状態」 （i アプリ起動中は設定不可）
 - マナーモードOFF
 - マナーモードON

5.39.2. オプションキーウィンドウの操作

端末状態ウィンドウは、エミュレーション端末の端末状態ウィンドウ表示を押すと表示されます（5.39. エミュレーション端末の操作 - ⑧ 参照）。



図 5-8 2 オプションキーウィンドウ

オプションキーウィンドウでは、エミュレーション端末ウィンドウにないキーのイベントを発生させます。

エミュレーション端末ウィンドウにないキーのイベントは以下のとおりです。

- 「ロールキー」
 - 右回転
 - 左回転
- 「背面キー」
 - 背面キー1
 - 背面キー2
 - 背面キー3
- 「各種キー」
 - カレンダーキー
 - ドライブモードキー
 - GPS
 - iモードキー
 - Myセレクトキー
 - メニューキー

- 「カメラ機能」
 - ズームイン
 - カメラキー
 - ズームアウト
 - ライトフラッシュ
 - シャッター
 - カメラセレクトキー

※ アプリケーション実行中、GPSキーでの測位はおこなわれません。

5.39.3. 3Dサウンド表示

3Dサウンド表示ウィンドウは、エミュレーション端末の3Dサウンド表示を押すと表示されます（5.39. エミュレーション端末の操作 – ⑨ 参照）。

3Dサウンドの表示は、サウンド設定が「ビジュアルモード」に設定されている必要があります（「5.6. サウンド設定」 参照）。「通常再生モード」では表示されませんのでご注意ください。

図 5-8 3 を用いて3Dサウンド表示ウィンドウの基本操作を説明します。

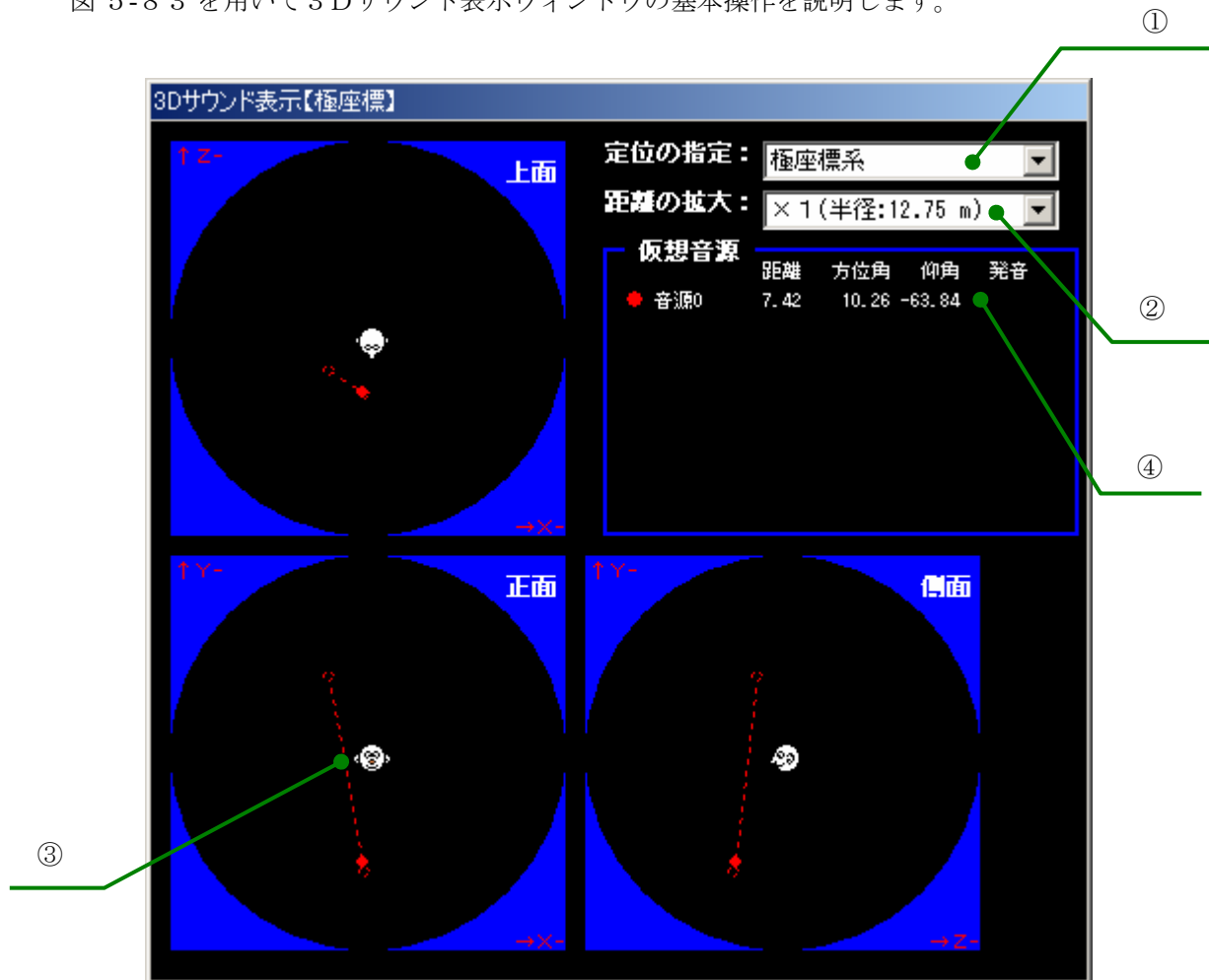


図 5-8 3 3Dサウンド表示

3Dサウンド表示ウィンドウでは、上、正面、側面の三面図で3Dの座標位置を表します。

- ① 定位の指定
「極座標」か「デカルト座標」を選択します。
- ② 距離の拡大
描画する半径の大きさを指定します。
(最大で半径 12.75m、最小で半径 1.27m で表示させることができます。)

③ 音源の移動

現在再生中の音源の移動位置が表示されます。

複数再生している場合は、複数の音源が表示されます。

定位が移動開始する始点、及び定位が移動完了する終点は点線の円で描画されます。

現在の定位は、塗りつぶされた円で表示され、移動の軌跡は点線で描画されます。

④ 仮想音源（情報）

仮想音源の定位情報を表示します。①の「定位の指定」により選択された座標系で、現在の定位情報を表示します。

「発音」の表示は MFi に埋め込まれたオーディオデータが再生中の場合は ♪ が表示されます。

停止中の場合は何も表示されません（アプリから 3D 環境を制御する場合、「発音」は何も表示されません）。

第6章. ツール

この章では、エミュレータで扱える便利なツールについて説明します。

6.1. スクラッチパッド管理ツール

スクラッチパッド管理ツールは、i アプリのスクラッチパッドファイルを 16 進数でダンプして編集するためのツールです。

スクラッチパッド管理ツールを使用すれば、スクラッチパッドのデータを作成したり、i アプリからスクラッチパッドにデータが適切に書き込まれているかを確認したりすることができます。

6.1.1. スクラッチパッドファイル

スクラッチパッドは、実際の端末では不揮発性の記憶領域ですが、エミュレータでは 1 つのファイルです。スクラッチパッドには、文字列以外に i アプリで使用可能な基本タイプのデータを格納できるため、通常テキストビューアなどで表示することができません。

エミュレータは、スクラッチパッドファイルを以下のディレクトリに格納します:

“C:\iDKDoJa5.0\apps\<プロジェクト名>\sp”

スクラッチパッドファイルは <プロジェクト名>.sp です。

スクラッチパッドの指定方法については、ADFのSPsize（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」）を参照してください。

6.1.2. スクラッチパッド管理ツールの起動

スクラッチパッド管理ツールを起動するには、

1. メニューから「ツール」→「スクラッチパッド...」を選択します。

「スクラッチパッド管理ツール」画面が表示されます（図 6-1）。



図 6-1 スラッチパッド管理ツール

2. 現在、読み込まれているプロジェクトの最初のスラッチパッドファイルの内容が、16進数で表示されます。
3. 値を変更したい場合は、変更したい所の値を選択して直接変更します。



4. 「前へ」ボタンを押すと、1つ前のスラッチパッドの内容が表示されます。
5. 「次へ」ボタンを押すと、1つ先のスラッチパッドの内容が表示されます。
6. プルダウンからスラッチパッドの連番を選択すると、そのスラッチパッドの内容が表示されます (図 6-2)。



図 6-2 スラッチパッドの選択

7. 「適用」ボタンを押すと、表示されているスクラッチパッドの変更がスクラッチパッドファイルに書き出されます。
8. 「取込」ボタンを押すと、カーソルの位置から外部のファイルを取り込むことができます。
「取込」ボタンを押すと、「ファイル選択」ダイアログが表示されるので、取り込みたいファイルを選択し、「開く」ボタンを押します。

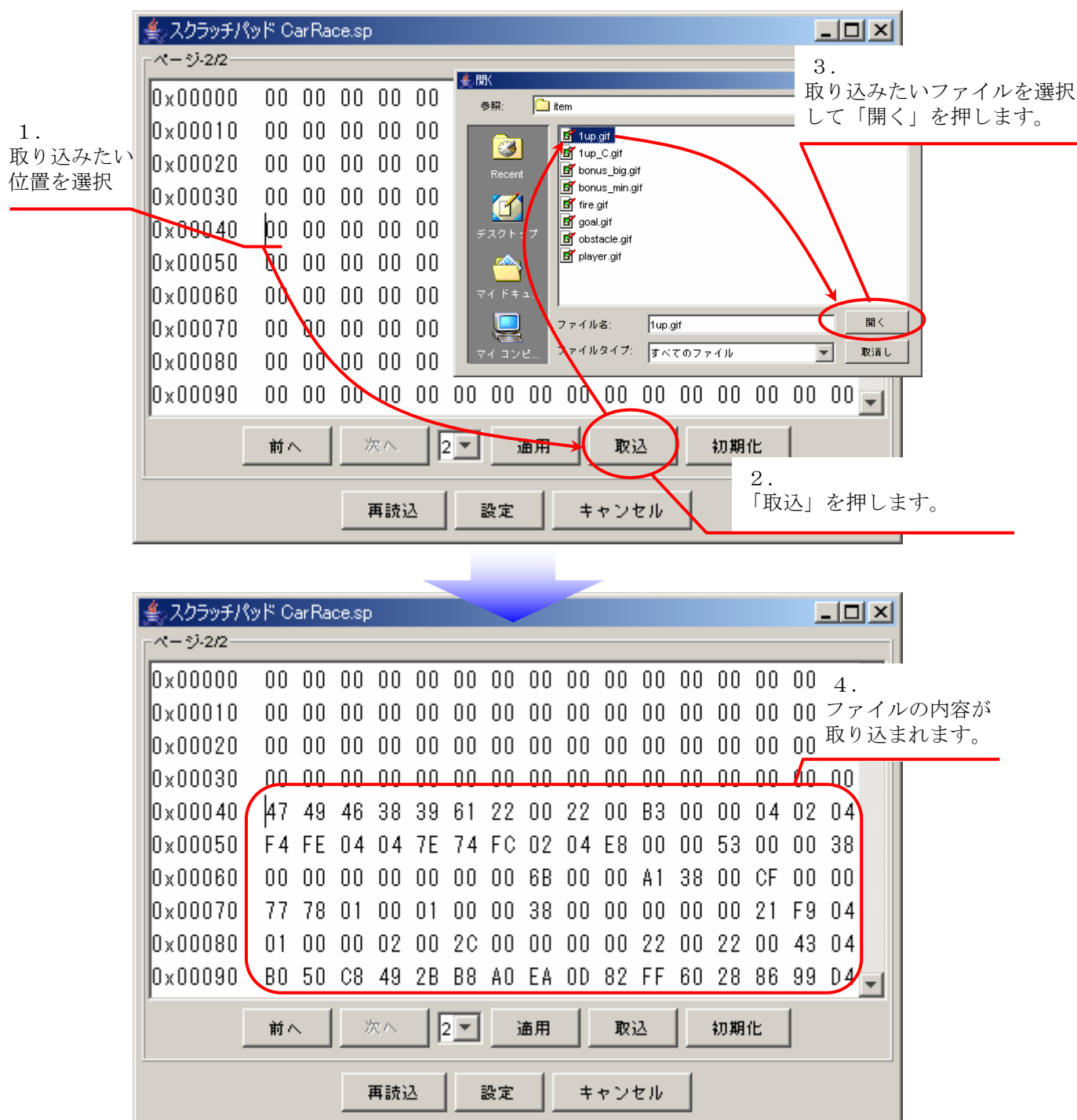


図 6-3 スクラッチパッドの置換処理

これにより、これまで他のバイナリエディタや HTTP サーバを経由してアプリケーションで取り込まないと作れなかったスクラッチパッドファイルをあらかじめ作成しておくことができます。

【注意】

スクラッチパッドのサイズは変わりませんので、取り込むファイルのサイズがスクラッチパッドのサイズを超えた場合は、ファイルの途中で切られてしまいます。その場合、全て取り込めなかった旨が警告されます。

9. 「初期化」ボタンを押すと、現在、表示されているスクラッチパッドが0で初期化されます。
このとき、スクラッチパッドを初期化してもよいか確認のダイアログが表示されますので、「はい」を選択します。



図 6-4 スクラッチパッド削除確認ダイアログ

10. 「再読込」ボタンを押すと、スクラッチパッドファイルの内容を全て再読込みし、スクラッチパッド管理ツールの表示時点もしくは「更新」を押した時点の状態に戻ります。
11. 「設定」ボタンを押すと、これまでの変更がスクラッチパッドファイルに書き出されます。
12. 「閉じる」ボタンを押すとスクラッチパッド管理ツールを終了します。

6.1.3. スクラッチパッドでJarファイルを扱う場合の注意事項

スクラッチパッドにJarを保存してアプリケーションで扱う場合、JarファイルのサイズをURLのlengthで指定するか、スクラッチパッドのサイズ（ADFのSPsize（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照））とJarファイルのサイズが同じでないとフォーマットエラーになります。

6.2. スクラッチパッドを全て初期化

メニューから「プロジェクト」→「スクラッチパッドを全て初期化」を選択することで、現在読み込まれているプロジェクトのスクラッチパッドを全て0に初期化することができます。このとき、スクラッチパッドをすべて初期化してもよいか確認のダイアログが表示されますので、「はい」を選択します。

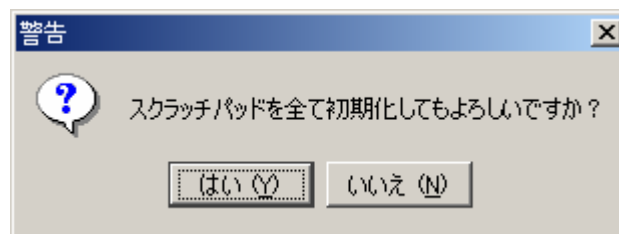


図 6-5 スクラッチパッド削除確認ダイアログ（全て）

スクラッチパッドが存在しない場合は、「スクラッチパッドがありません」というダイアログが表示されます。

6.3. 仮想 FeliCa データファイルビューワ

仮想 FeliCa データファイルビューワは、仮想 FeliCa データファイルを 16 進数でダンプします。仮想 FeliCa データファイルビューワを使用すれば、i アプリから仮想 FeliCa データファイルにデータが適切に書き込まれているかどうかを確認することができます。

6.3.1. 仮想 FeliCa データファイル

仮想 FeliCa データファイルは、以下のディレクトリに作成します：

C:\iDKDoJa5.0\apps\<プロジェクト名>\<プロジェクト名>.fc

6.3.2. 仮想 FeliCa データファイルビューワの起動

仮想 FeliCa データファイルビューワを表示するには、

1. メニューから「ツール」→「仮想FeliCaデータファイルビューワ...」を選択します。

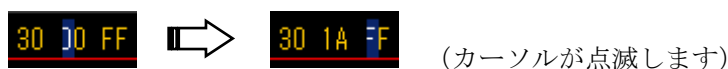
「仮想FeliCaデータファイルビューワ」画面が表示されます（図 6-6）。



図 6-6 仮想 FeliCa データファイルビューワ画面

インストール直後のデータファイルは、データ領域を除き、適切な値が入っています。仮想FeliCaデータファイルの構造については、「10.3.2. 仮想FeliCaデータファイル」を参照してください。

2. 現在読み込まれているプロジェクトの仮想 FeliCa データファイルの内容が 16 進数で表示されます。
3. 上段のプルダウン（※）から仮想 FeliCa データファイルの記憶領域を選択します。該当する記憶領域のデータの色が変わります。
これにより、実際に i アプリから書き込まれた領域を参照する事ができます。
4. 値を変更したい場合は、変更したい所の値を選択し直接変更します。
仮想 FeliCa データファイルビューワでは、CPID Data Block, Data Block, PIN Data Block のみ値の変更が可能です。



- Area DefineBlock, Service Block 1, Service Block 2, PIN Service Block は変更しても無効です。
5. 「更新」ボタンを押すと、これまでの変更結果が仮想 FeliCa データファイルに書き出されます。
 6. 「再読み込み」ボタンを押すと、仮想 FeliCa データファイルの内容を再度読み込み、仮想 FeliCa データファイルビューワの表示時点もしくは「更新」を押した時点の状態に戻ります。
 7. 「閉じる」ボタンを押すと仮想 FeliCa データファイルビューワを終了します。

仮想 FeliCa データファイルビューワで表示されたフリー領域を扱う i アプリを利用するには、次の条件が満たされている必要があります：

- CPID Data Blockが、ADFのFeliCaCPIDと同じ値でなければなりません（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。
- PIN Data Blockの先頭4バイトが、ADFのFeliCaPINの値を16進数表現した値でなければなりません（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）。

これは、ADFの設定で FeliCaCPID、FeliCaPIN を設定した場合に自動的に関連付けをおこなう事ができます（「4.7.2. FeliCaフリー領域の関連付け」参考）。

手動で設定するには：

1. 仮想 FeliCa データファイルビューワの上段にあるプルダウンから CPID Data Block を選択します。
2. CPID を格納する場所が桃色で表示されるので、この部分にカーソルを当て、ADF に設定した FeliCaCPID と同じ値に変更します。
3. 仮想 FeliCa データファイルビューワの上段にあるプルダウンから PIN Data Block を選択します。

4. PIN を格納する場所が紫色で表示されるので、この部分にカーソルを当て、先頭 4 バイトに ADF に設定した FeliCaPIN の値を 16 進数表現で設定します。例えば、ADF に設定した FeliCaPIN の値が 12345678 の場合は、PIN Data Block の先頭 4 バイトを 00BC614E に変更します。
5. 仮想 FeliCa データファイルビューワの「更新」ボタンを押します。
6. 「閉じる」ボタンを押します。

【注意事項】

- 仮想 FeliCa データファイルビューワを表示している状態で、i アプリから仮想 FeliCa データファイルヘータを書き込んだ場合は、「再読込」ボタンを押してください。
- i アプリが起動している状態では、仮想 FeliCa データファイルビューワから値を変更しないでください。もし、変更した場合は i アプリが不正な動作をすることがあります。
- フリー領域を利用する i アプリで、ADF の FeliCaCPID、FeliCaPIN の値を設定して、仮想 FeliCa データファイルを作成しても、フリー領域の CPID Data Block、PIN Data Block には自動的に反映されません。

6.3.3. 仮想 FeliCa データファイルビューワの表示色

仮想 FeliCa データファイルビューワの表示で、ウィンドウ内の色は下記のようになっています。

- 橙 : System Define Block
- 緑 : Area Define Block
- 赤 : Service Block
- 水色 : PIN Service Block
- 桃 : CPID Data Block および Data Block
- 紫 : PIN Data Block
- 灰色 : 未使用ブロック

6.3.4. ブロックの構成

仮想 FeliCa データファイルビューワでは各ブロックは下記のような構成になっています。

1 ブロックは 16 バイトで、以下のようなバイト列になっています。

D0	D1	D2	D3	・ ・ ・ ・ ・	D13	D14	D15
----	----	----	----	-----------	-----	-----	-----

D = バイト

DoJa-4.1(以前)用の場合

- ・ SystemDefineBlock
D0～D1 : System Code
- ・ AreaDefineBlock
D0～D1 : Area Code
D2～D3 : End Service Code
- ・ Service Block
D0～D1 : Service Code

DoJa-5.0 用 の場合

- ・ SystemDefineBlock
D0～D1 : System Code
- ・ AreaDefineBlock
D0～D3 : Area Code
D6～D9 : End Service Code
- ・ Service Block
D0～D3 : Service Code

※ 扱う仮想FeliCaデータが DoJa-4.1 用 か DoJa-5.0 用 かは、仮想FeliCa設定のバージョンで変更できます（「5.1 0. 仮想FeliCa設定」参照）。

仮想 FeliCa データファイルビューワで編集を行う場合のエンディアン表記については、全てビッグエンディアンになっています。

6.4. Javadoc 作成ツール

Javadoc 作成ツールは、i アプリのドキュメンテーションコメントから javadoc を作成します。
i アプリライブラリのjavadocを作成する場合は、スタブソースを外してから実行してください（i アプリライブラリについては、「6.7. i アプリライブラリ」参照）。

Javadoc 作成ツールで javadoc を作成するには：

1. メニューから「ツール」→「javadoc作成ツール...」を選択します（図 6-7）。

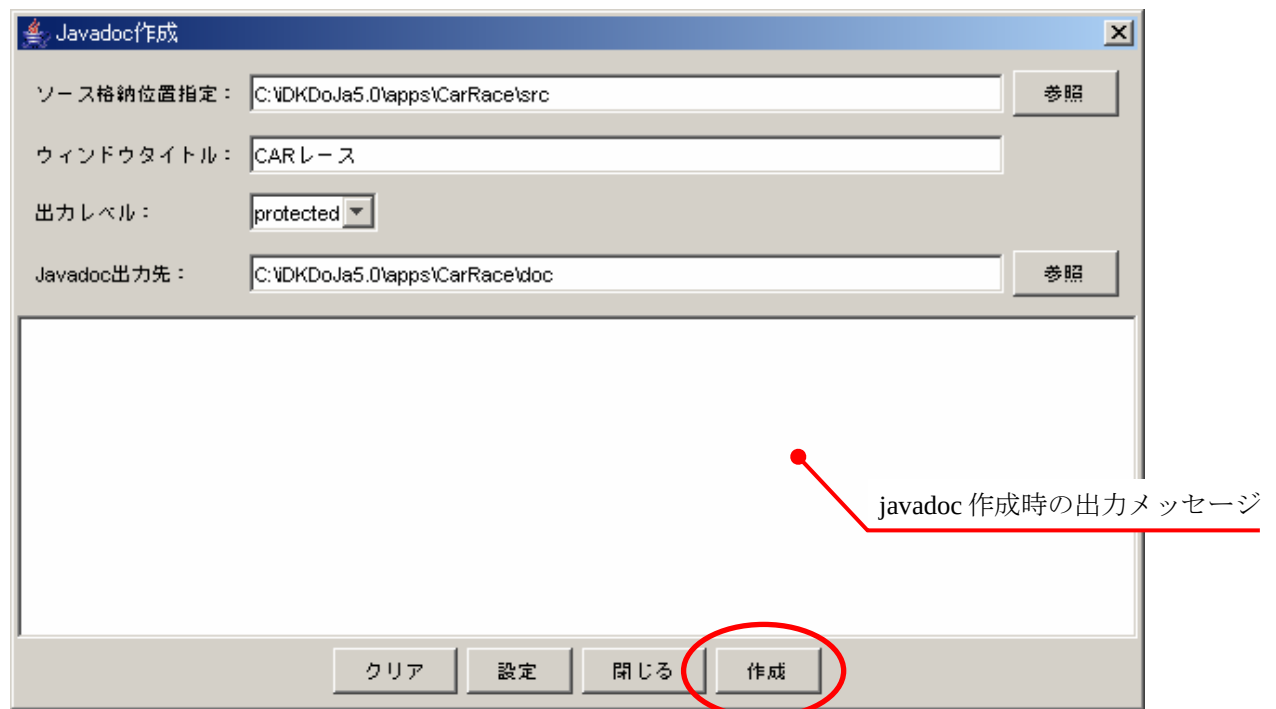


図 6-7 javadoc 作成ツール

2. 各設定項目を入力して「作成」ボタンを押すと javadoc が作成されます。
まずは、サンプルのCarRaceで試してみてください。
各設定項目は表 6-1 に示すとおりです。

表 6-1 ヘルプの設定項目

設定項目	説明	デフォルト
ソース格納位置指定	i アプリのソースが置かれたディレクトリ	読み込まれたプロジェクト
ウィンドウタイトル	javadoc のタイトル	読み込まれたプロジェクトの ADF の AppName
出力レベル	javadoc で出力するスコープ	protected
Javadoc 出力先	javadoc の出力先ディレクトリ	読み込まれたプロジェクト直下に doc で作成

3. 「設定」 ボタンを押すと設定を保存します。

設定された状態は、次のファイルに保存されています。

<プロジェクト名>\javadoc.properties

「javadoc.properties」 ファイルの記述方法は、以下のとおりです：

項目	内容
srcdir = C:\iDKDoJa5.0\apps\CarRace\src	ソース格納位置
windowtitle = CAR レース	ウィンドウタイトル
d = C:\iDKDoJa5.0\apps\CarRace\doc	javadoc 出力先
xxx = xxx	javadoc のパラメータで値指定のあるもの (※) <例> encoding = "Shift_JIS" footer = "フッター" ただし、以下のパラメータは扱えません。 • help • doclet <class> • docletpath <path> • breakiterator • source <release> • extdirs <dirlist> • verbose • J<flag> • docfilessubdirs • link • linkoffline • excludedocfilessubdir • quiet • serialwarn • taglet • tagletpath • stylesheetfile • bootclasspath <pathlist> • classpath <pathlist> bootclasspath、classpath は内部処理で作成するため、ユーザ指定は対応しません。
xxx	javadoc のパラメータで値指定のないもの (※) <例> private use

【注意】

本機能で設定可能なオプションは j2sdk1.4.2_04 に含まれる javadoc でサポートされている範囲のものであり（上記に記載されている対象外パラメータは除く）、それ以降に追加されたオプションについてはサポートしておりません。（無効なオプションとして処理されます。）

※ javadoc の各パラメータにつきましては、下記方針に従ってチェックを実施しております。これにあわせて指定して下さい。

<形式チェック方針>

- 1) 値指定が必要なパラメータに対しては、値が設定されていること。
また、値指定が不要なパラメータに対しては、値が設定されていないこと。
- 2) ファイルを指定する場合は、指定ファイルが存在すること。
また、パスを指定する場合は、指定パスが存在すること。
- 3) javadoc に文字を埋め込むようなパラメータの値、及び encoding, docencoding, charset の値は""で囲むこと。
上記以外のパラメータの値は""で囲まないこと。

6.5. ADFチェックツール

ADF チェックツールは、H T T Pサーバ上に配置された i アプリの ADF ファイル (.jam) が正しく作成されているかをチェックするためのツールです。

実際の i モード携帯電話機では、A D F が間違っているにもかかわらず単純なエラーメッセージしか表示されず、何が間違っているかが分からないケースがあります。
本ツールでは、こうした現象を解消するため、H T T Pサーバ上に設置された i アプリの A D F ファイルを読み取ってチェックする事ができます。

A D F チェックツールを起動するには：

1. メニューから「ツール」→「A D F チェック...」を選択します（図 6-8）。

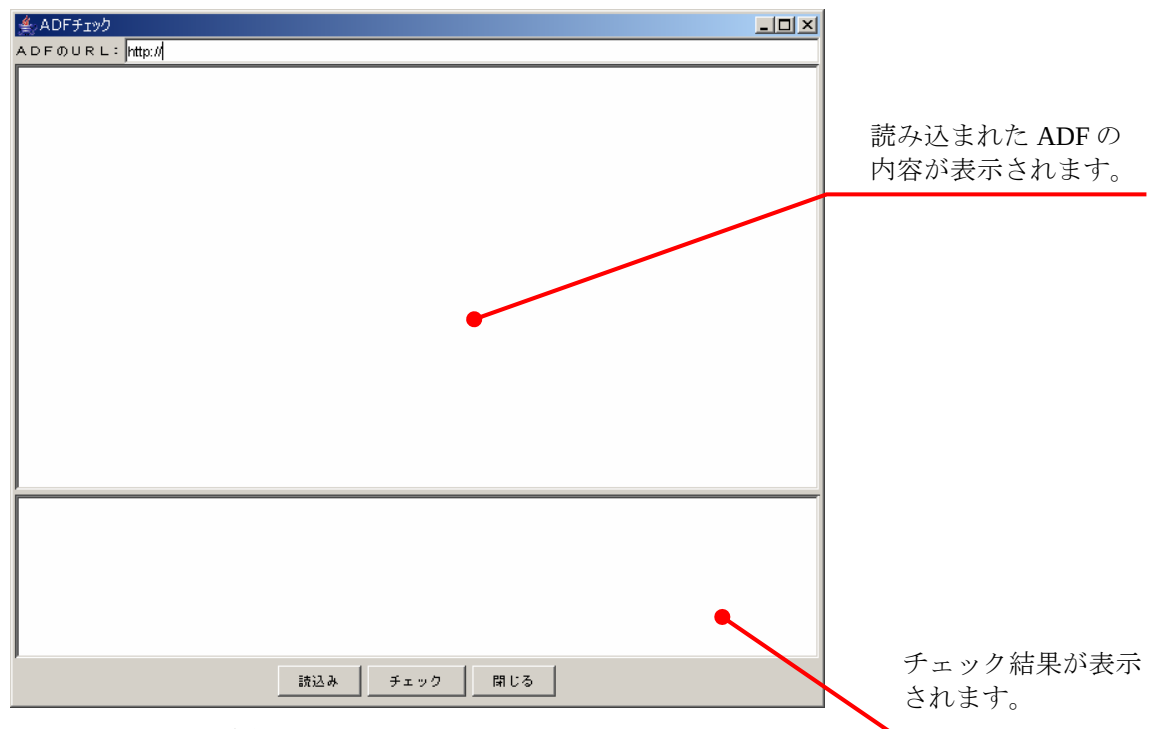


図 6-8 ADFチェックツール

2. 「ADFのURL」にチェックしたいADFのURLを指定します。
このとき、「file:ADFへのパス」を指定するとローカルにあるADFをチェックすることもできます。（例：file:c:/iDKDoJa5.0/apps/CarRace/bin/CarRace.jam）
3. 「読み込み」ボタンを押すとADFをH T T P経由で取得します。
4. 「チェック」ボタンを押すと読み込んだADFの構文が正しいかどうかをチェックします（図 6-9）。

「ADFのチェックは正常に終了しました」と表示されれば、ADFは正しいです。

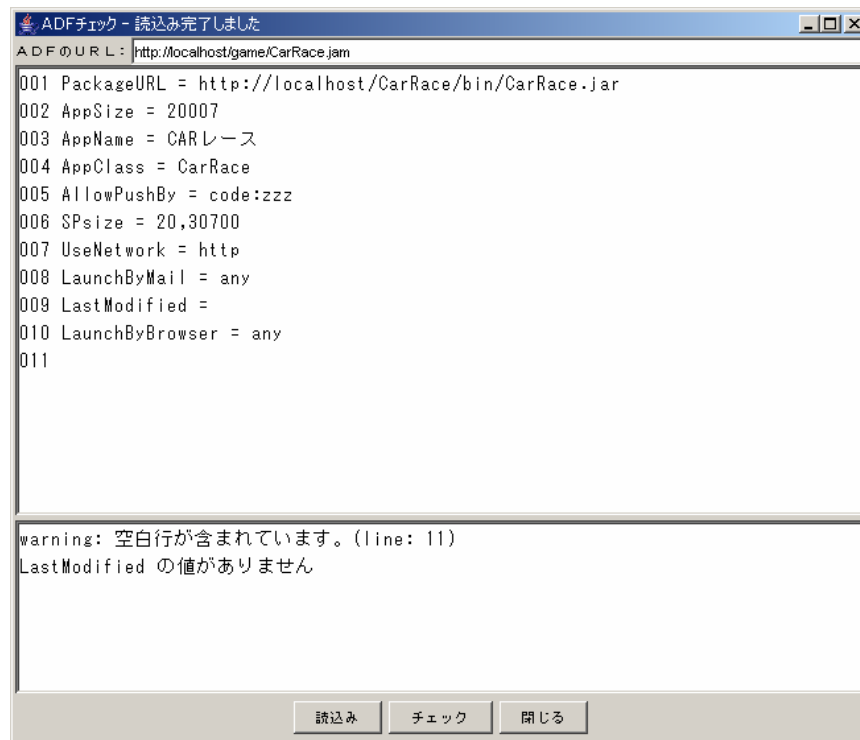


図 6-9 ADF のチェック実行

- ※ HTTP サーバで認証処理がある場合、ダウンロードする事ができない場合があります。
- ※ Jar との整合性はチェックしていませんので、もし、本ツールで正しくても i モード携帯電話上で実行できない場合は、PackageURL や AppClass が正しいかどうかを確認してみてください。

ワンポイント

ADFにはiアプリ(Jar)のサイズの指定(AppSize)がありますので、もし、iアプリをビルドしなおした場合はADFも更新されています。
そのため、Jarに変更があった場合は、必ずADFも置き換えるようにしてください。
(Jarのみの置き換えはNGと考えましょう。)

6.6. スクリーンショット

スクリーンショットは、端末ウィンドウの i アプリ実行画面をクリップボードもしくはファイルに保存する機能です。（クリップボードとは、Windows のカット＆ペーストをする作業領域です。）

スクリーンショットを実行するには：

1. i アプリを実行します。
2. スクリーンショットを取りたい時に、メニューから「ツール」→「スクリーンショット」を選択します（図 6-10）。
（あらかじめ、ショートカットキーを設定しておくとう便利です（「5.2.8. ショートカットキーの設定」参照）。）

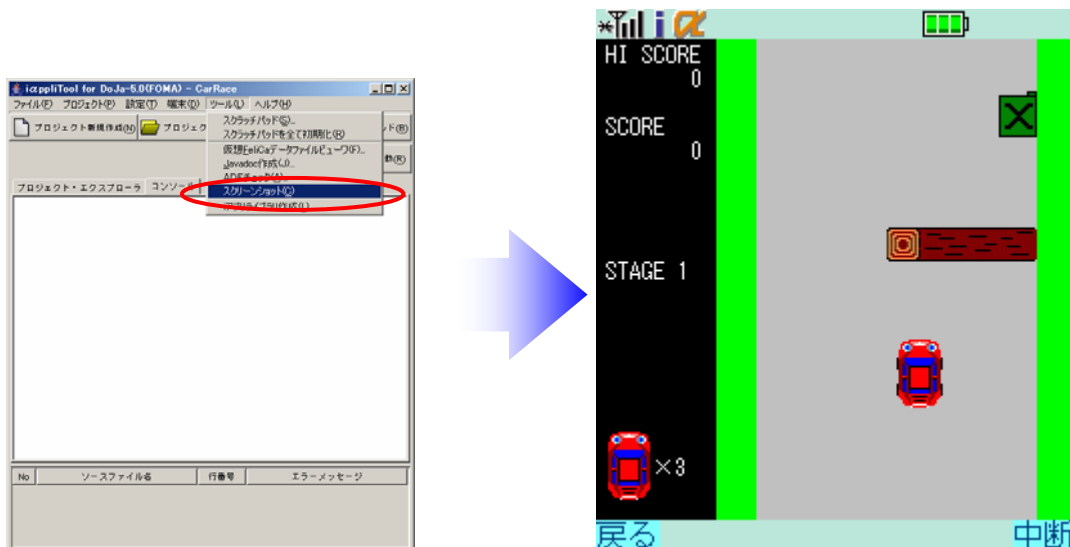


図 6-10 スクリーンショットの実行

3. クリップボードもしくはファイルに保存されているので、これを i アプリの説明などに利用します。

スクリーンショットをどこに保存するかは、「5.3.3. スクリーンショット保存の設定」を参考にしてください。

6.7. iアプリライブラリ

iアプリを作成すると同じような処理が多く存在してきます。
本機能では、これらの同じような処理をiアプリライブラリとしてまとめ、他のiアプリで利用できるようにします。

6.7.1. iアプリライブラリの作成

iアプリライブラリを作成するには：

1. まず、ライブラリ用のプロジェクトを作成します。
メニューから「プロジェクト」→「新規作成」を選択します（図 6-1 1）。

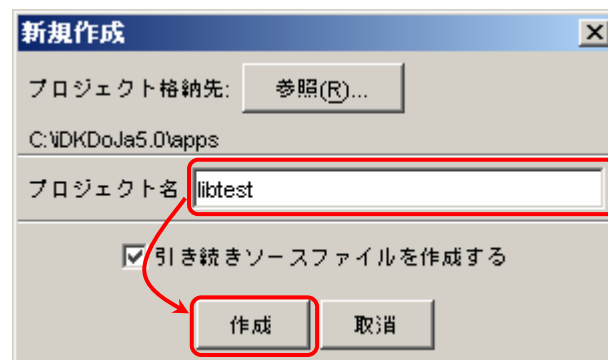


図 6-1 1 iアプリライブラリ プロジェクト新規作成

2. プロジェクト名を入力して「作成」を押します。
3. 次にメニューから「プロジェクト」→「ソースファイルの新規作成」を選択して、ソースファイルを作成します（図 6-1 2）。

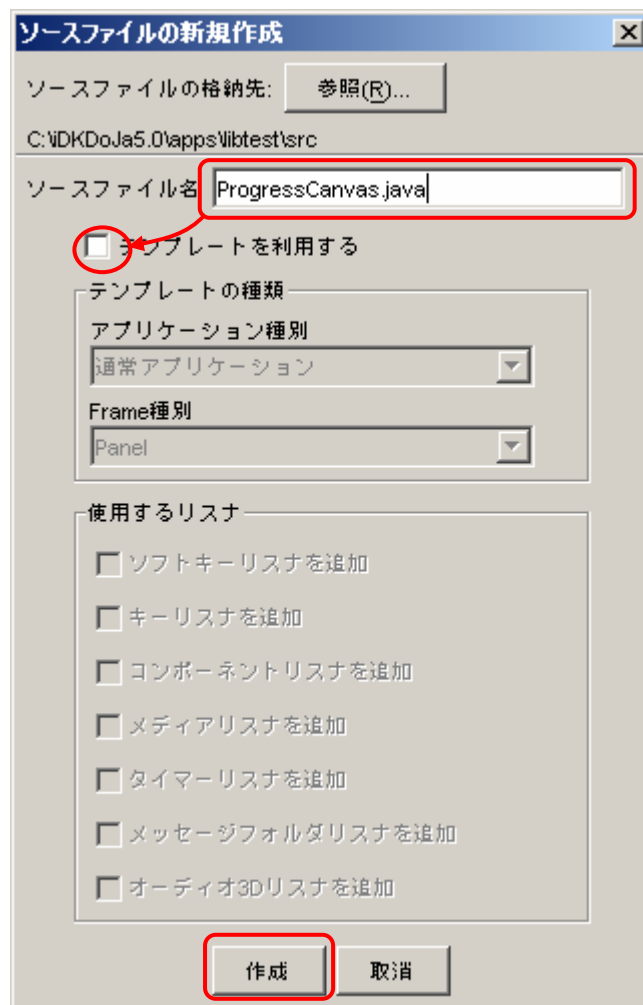


図 6-1 2 i アプリライブラリ ソースファイル新規作成

4. ソースファイル名を入力して、「作成」を押します。
(ここでは、「テンプレートを利用する」のチェックを外しておきます)

5. 開かれたソースファイルに共通処理のプログラムを書き込みます。

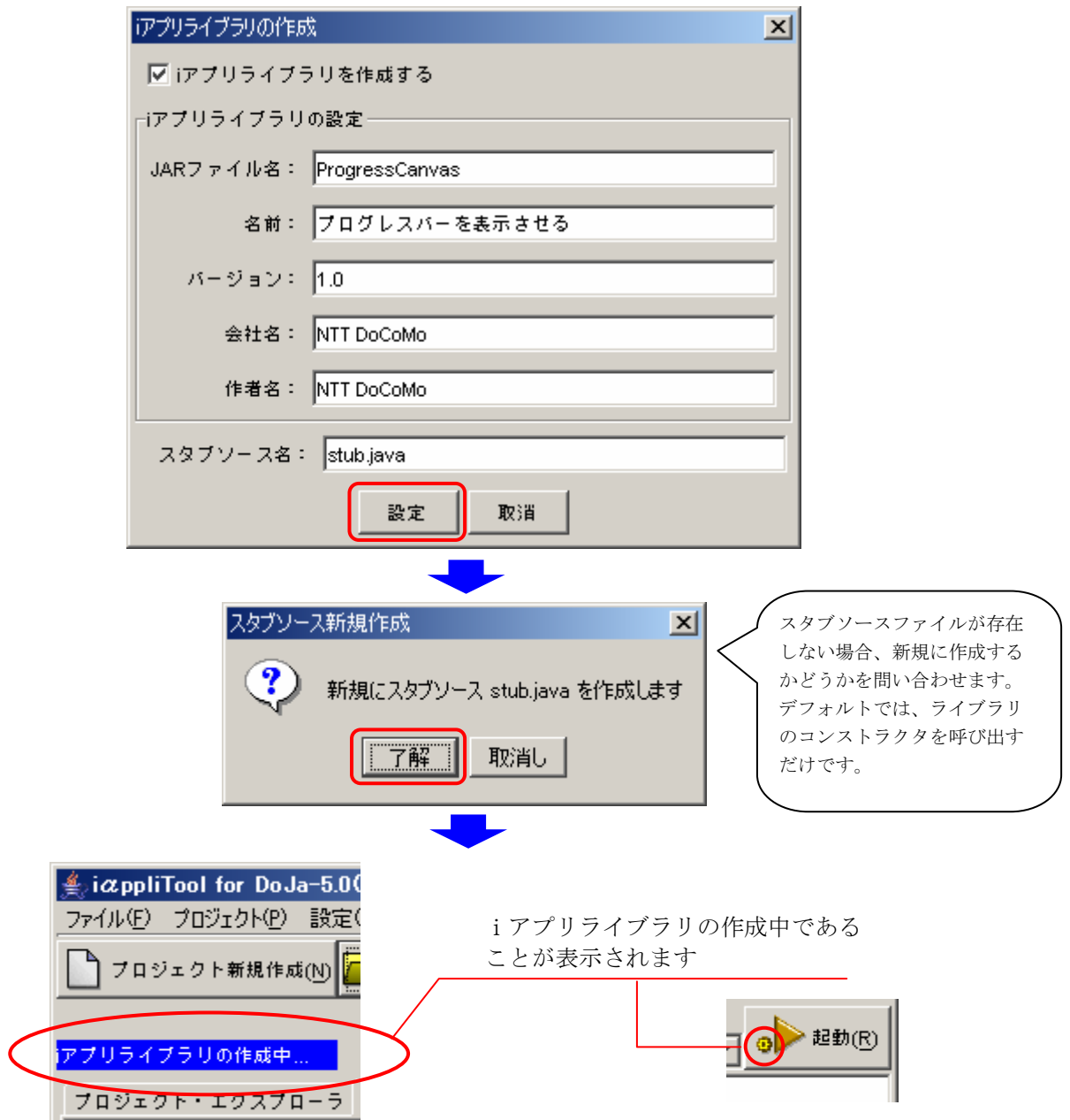


図 6-13 i アプリライブラリ プログラムの作成

(図の例は、サンプルアプリ CarRace の ProgressCanvas.java を書き換えたものです。)

6. 「ツール」 → 「i アプリライブラリ作成」を選択します。

7. 「i アプリライブラリを作成する」にチェックをして、このライブラリの属性「Jarファイル名、名前、バージョン、会社名、作者名」と「スタブソース名」(.java)を設定します(図 6-1 4)。



ここで設定した属性は、i アプリライブラリ中の Jar に META 情報で格納されます。
i アプリライブラリを使う i アプリや i アプリライブラリそのものには影響ありません。

各設定項目は表 6-2 に示すとおりです。

表 6-2 i アプリライブラリの設定項目

設定項目	説明	デフォルト
Jar ファイル名	i アプリライブラリの Jar ファイル名	なし
名前	i アプリライブラリの名前 (省略可)	なし
バージョン	ライブラリのバージョン番号 (省略可)	なし
会社名	作成者の会社名 (省略可)	なし
作者名	作者名 (省略可)	なし
スタブソース名	ライブラリをテストするためのプログラムです。あらかじめ作成しておかなくてもテンプレート ^[9] から自動的に作成されます (図 6-15)。 省略すると”プロジェクト名_stub.java”が作成されます。	xxx_stub.java

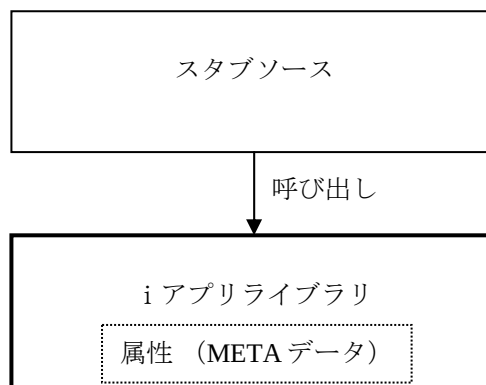


図 6-15 スタブソースと i アプリライブラリの関係

^[9] スタブソースのテンプレートは、以下のファイルに格納されています。
 ファイル中の"{CLASSNAME}"がクラス名、"{AUTHOR}"が作者名、"{VERSION}"がバージョン、"{DATE}"が日時、"{COMP_NAME}"が名前に変更されます。

C:\iDKDoJa5.0\lib\ComponentStub.txt

8. 「プロジェクト」→「ソースファイルを開く」で、スタブソースを開き、ライブラリの呼び出しに合うようにプログラムを変更します。

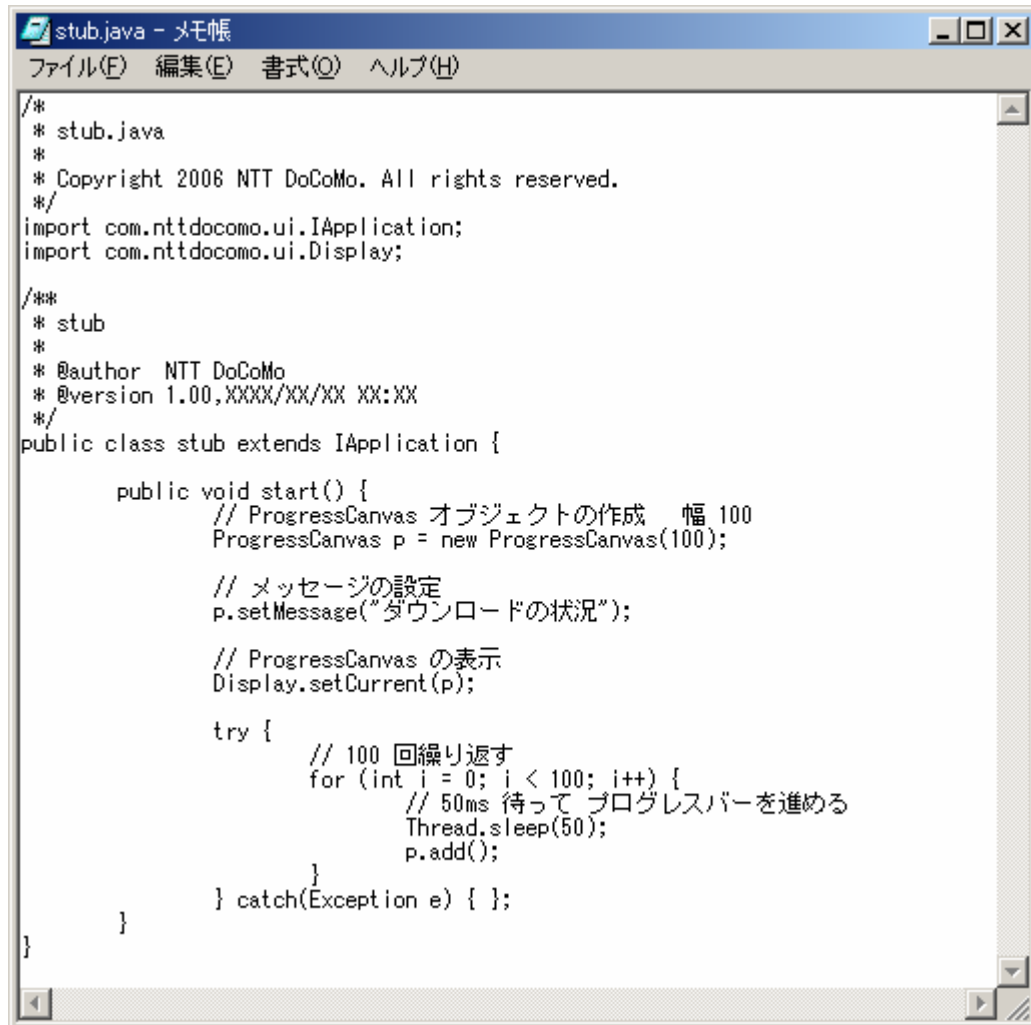


図 6-16 i アプリライブラリ スタブソースの書き換え

9. 「ビルド」を実行します。

10. 「起動」を押して i アプリライブラリを実行（テスト）します。

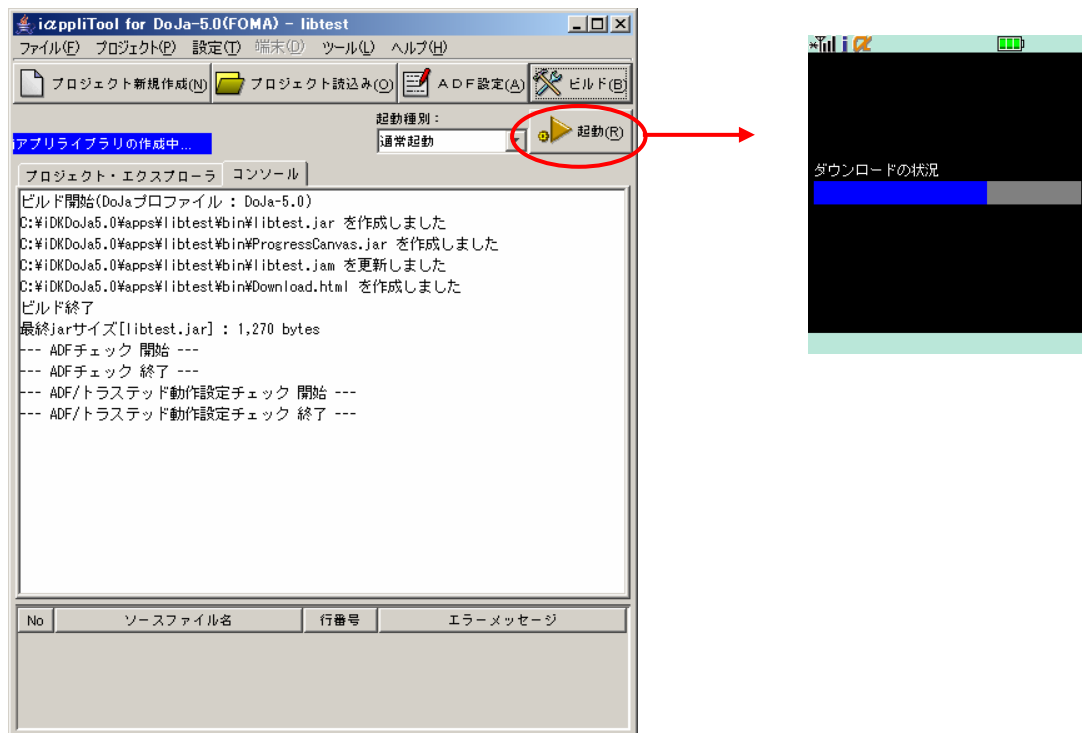


図 6-17 i アプリライブラリの実行

11. 次の場所に完成した i アプリライブラリができていますので配布して再利用します。

C:\iDKDoJa5.0\apps\プロジェクト名\bin\ライブラリ名.jar

注意：

スタブソースには、内部クラス（inner class）を作成しないでください。
もし内部クラスがある場合には、ライブラリ側に取り込まれてしまいます。
それでも実行上の問題はありませんが、不要なコードが含まれたライブラリになります。

旧エミュレータで作成された i アプリライブラリを本エミュレータで実行するには、必ず、ビルドしなおしてください。

6.7.2. iアプリライブラリの利用

i アプリライブラリを利用するには：

1. i アプリのプロジェクトを読み込みます。
2. メニューから「プロジェクト」→「i アプリライブラリ」を選択します。

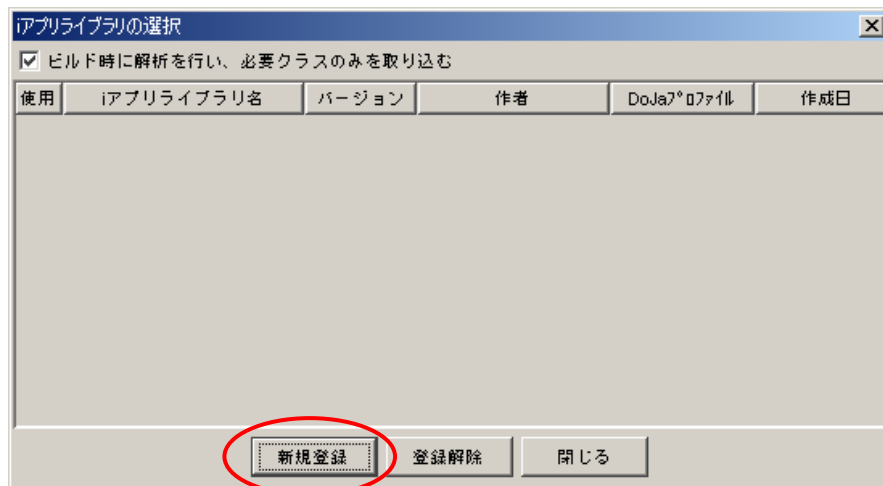


図 6-18 i アプリライブラリの選択画面

3. i アプリライブラリの選択画面が表示されるので「新規登録」を押します。
4. ファイル選択ダイアログで登録したいi アプリライブラリ(.jar)を選択するとi アプリライブラリの選択画面に追加されます。
ここで、登録されたライブラリは、他のプロジェクトでも利用できます。

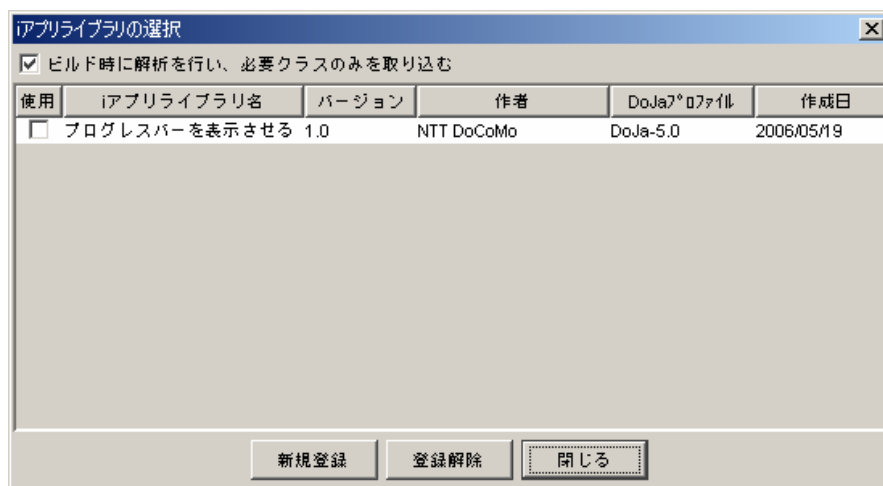


図 6-19 i アプリライブラリの選択画面 - ライブラリの追加

💡 ワンポイント

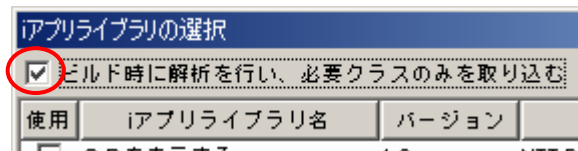
次の場所にi アプリライブラリを置いておくと自動的に登録できます。

C:\iDKDoJa5.0\lib\component

5. 作成した i アプリがライブラリ中の使われないクラスまで全て取り込んでしまうと Jar ファイルは膨大なサイズになってしまいます。これを防ぐために「ビルド時に解析を行い、必要なクラスのみを取り込む」にチェックを付けると、i アプリのビルド時にライブラリ中の(使われている)必要なクラスだけが Jar に組み込まれるようになります。

これはデフォルトでチェックが付いています。

もし、i アプリの実行が失敗するようであれば、このチェックを外してください。



6. 不要になった i アプリライブラリは、一覧から選択して「登録解除」を押します。
(※ C:\iDKDoJa5.0\lib\component のライブラリは解除できません。)
7. 登録されたライブラリから使いたいライブラリにチェックをつけます。

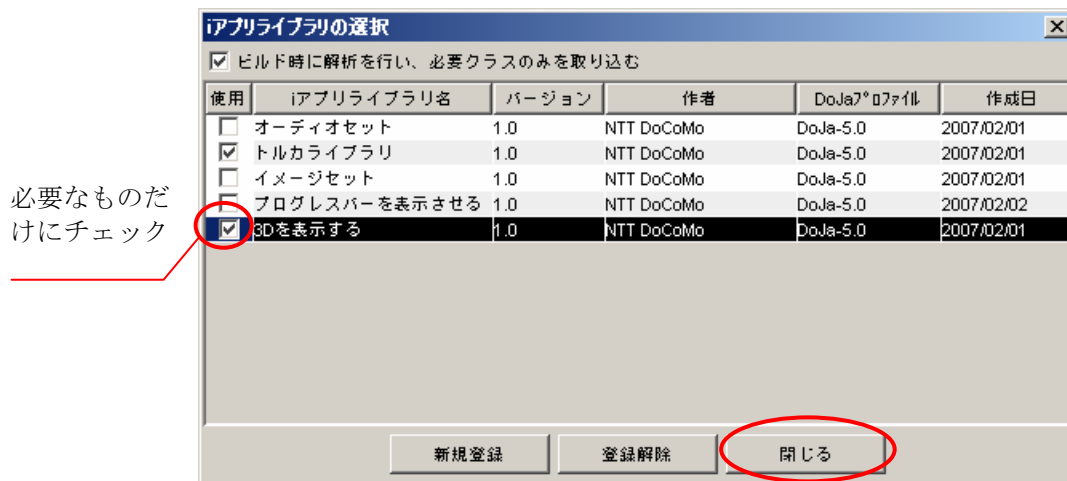


図 6-20 ライブラリの選択

8. 「閉じる」を押します。
9. i アプリをビルドします。
これだけで、チェックのついた i アプリライブラリが自動的に組み込まれ、i アプリからライブラリ中のクラスやメソッドを呼び出すことができますようになります。

【ご注意】

i アプリライブラリの登録では、i アプリライブラリの格納先が記憶されているので、i アプリライブラリを別の場所に移動した場合は、登録しなおす必要があります。

i アプリライブラリの作成で別の i アプリライブラリを利用する事も可能です。ただし、別の i アプリライブラリは作成中の i アプリライブラリには組み込まれません。

6.8. CPBindingAuth 署名付与ツール

i アプリから SD メモリなどのストレージデバイスにアクセスするためには、ADF の CPBindingAuth と PackageHashCode の設定が必要になります。（ADF については、「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照）

本ツールは、証明書と i アプリの Jar ファイルから CPBindingAuth キーと PackageHashCode キーを生成して ADF ファイルに追加します。

本ツールは、ドコモが発行するユーザ証明書をを使うもの (bindsigner) とユーザが独自に発行した証明書を使うもの (bindsigner2) の 2 通りがあります。

6.8.1. ドコモが発行したユーザ証明書を使って署名する方法

USB ケーブルで繋いだ FOMA 端末からドコモが発行するユーザ証明書を使って、ADF の CPBindingAuth と PackageHashCode を設定します。

ドコモが発行するユーザ証明書は、FOMA 向けのクライアント認証サービス **FirstPass** で携帯端末上に取得することができます。

CPBindingAuth 署名付与ツールの概要図を図 6-2 1 に示します。

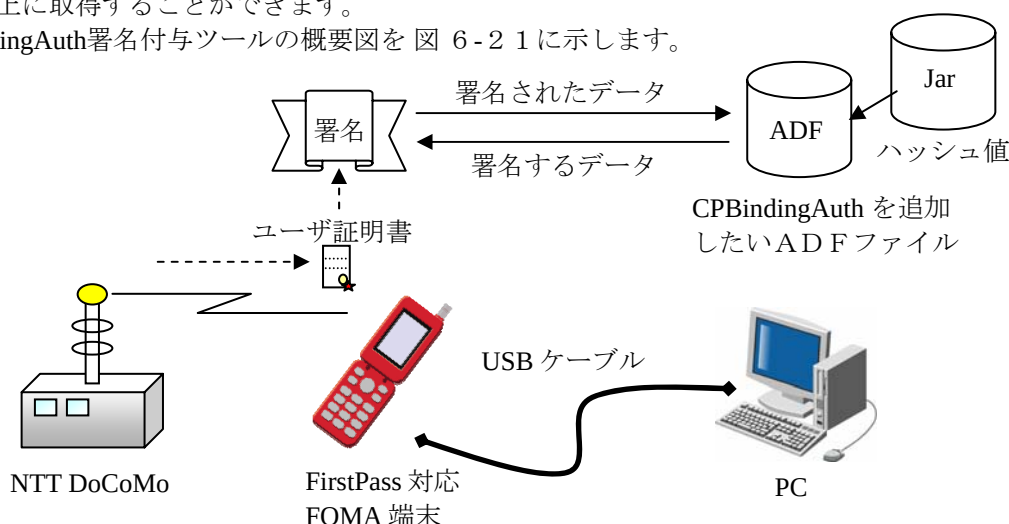


図 6-2 1 CPBindingAuth 署名付与ツール 概念図

CPBindingAuth を設定するには：

1. FOMA 端末で i モードメニューから FirstPass のユーザ証明書を取得しておきます。
2. FOMA 端末と PC を USB ケーブルで接続します。
USB ケーブルを接続するには、別途、機種毎にドライバが必要となります。
各ドライバは、<http://www.nttdocomo.co.jp/service/data/foma/> → 「通信設定ファイル」から入手できます。
3. コマンドプロンプトを開き、C:\iDKDoJa5.0\bin に移動します。
4. CPBindingAuth 署名付与ツールを実行します。
ツールの実行形式は以下のとおりです。

```
bindsigner -adf ADF -jar JAR -pin2 PIN2_CODE -port COM_PORT | [-help]
```

```
ADF          : CPBindingAuth キーを追加したい A D F ファイル
               (UseStorage が設定されている必要があります。)
JAR          : i アプリの J a r ファイル
PIN2_CODE    : F O M A 端末の P I N 2 コード
COM_PORT     : U S B ケーブルを接続している C O M ポート名
               (Windows のデバイスマネージャで確認してください。)
```

<実行例>

```
bindsigner -adf CarRace.jam -jar CarRace.jar -pin2 0000 -port COM1
```

5. 「xxx.jam の CPBindingAuth に署名しました。」と表示されたら、ADF ファイルに CPBindingAuth と PackageHashCode が追加されています。
このとき既に A D F ファイルに CPBindingAuth や PackageHashCode が存在する場合は書き換えられます。

6.8.2. ユーザが発行した証明書を使って署名する方法

ユーザが発行した証明書を使って、A D F に CPBingingAuth の署名データを設定します。
この証明書は、**PKCS#12 形式**で秘密鍵が格納されていなければなりません。

CPBindingAuth を設定するには：

1. 認証用の PKCS#12 ファイルを用意します。
2. コマンドプロンプトを開き、C:\iDKDoJa5.0\bin に移動します。
3. CPBindingAuth 署名付与ツールを実行します。
ツールの実行形式は以下のとおりです。

```
bindsigner2 -adf ADF -jar JAR -p12 PKCS#12 | [-help]
```

```
ADF          : CPBindingAuth キーを追加したい A D F ファイル
               (UseStorage が設定されている必要があります。)
JAR          : i アプリの J a r ファイル
PKCS#12      : P K C S # 1 2 ファイル
```

<実行例>

```
bindsigner2 -adf CarRace.jam -jar CarRace.jar -p12 usercert.p12
```

4. 「Enter Password:」で、PKCS#12 ファイルに設定されたパスワードの入力を聞いてく
るので、パスワードを入力します。
5. 「xxx.jam の CPBindingAuth に署名しました。」と表示されたら、ADF ファイルに CPBindingAuth と PackageHashCode が追加されています。
このとき既に A D F ファイルに CPBindingAuth や PackageHashCode が存在する場合は書き換えられます。

第7章. エミュレータの終了

エミュレータを終了するには、メニューから「ファイル」→「終了」を選択します。
エミュレータを終了してもエミュレータに設定した内容は消えません。また、一部の設定は、Windows ユーザ毎に保持されています。

第8章. デバッグトレース

高度な使い方として、「C:\iDKDoJa5.0\bin\doja_g.exe」を実行する事により、コンソールにKVMのデバッグトレースを出力する事ができます。^[10]

デバッグトレースを実行するには：

1. doja_g.exe を実行し、メニューから「設定」→「デバッグトレース設定...」を起動します（図 8-1）。

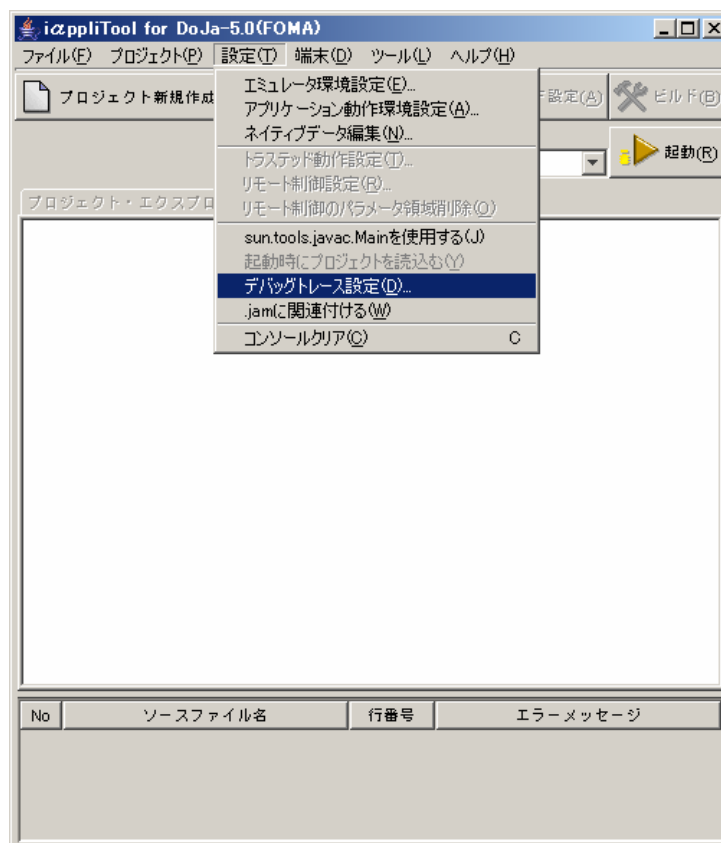


図 8-1 デバッグトレースの設定

[10]

doja_g.exe でビルドした i アプリは、例外の発生などでより詳細な情報（クラスやメソッド）を出力します。
ただし、デバッグ情報が含まれるため、Jar ファイルのサイズが大きくなってしまいます。
デバッグ作業が終わったら、必ず正規の doja.exe でビルドしなおしてください。

2. デバッグトレース設定画面（図 8-2）が表示されるので、デバッグトレースの出力と種類を選び、「設定」を押します。
選択できるデバッグトレースの出力は以下のとおりです。

- 「無効」 デバッグトレースを出力しません。
- 「有効（全て）」 全てのトレースを出力します。
- 「有効（選択）」 トレースの種類を選択できます。
複数選択したい場合は、[Ctrl]キーを押しながら選択してください。

【トレースの種類】

```

traceallocation.....trace memory allocation
tracegc.....trace garbage collection
tracegcverbose.....trace garbage collection, more verbose
traceclassloading.....trace class loading
traceclassloadingverbose.....trace class loading, more verbose
traceverifier.....trace class file verifier
tracestackmaps.....trace the behavior of stack maps
tracebytecodes.....trace bytecode execution
tracemethods.....trace method calls
tracemethodsverbose.....trace method calls, mode verbose
tracestackchunks.....trace the allocation of new stack chunks
traceframes.....trace stack frames
traceexceptions.....trace exception handling
traceevents.....trace the behavior of the event system
tracemonitors.....trace the behavior of monitor objects
tracethreading.....trace the behavior of the multithreading system
tracenetworking.....trace the network access
    
```

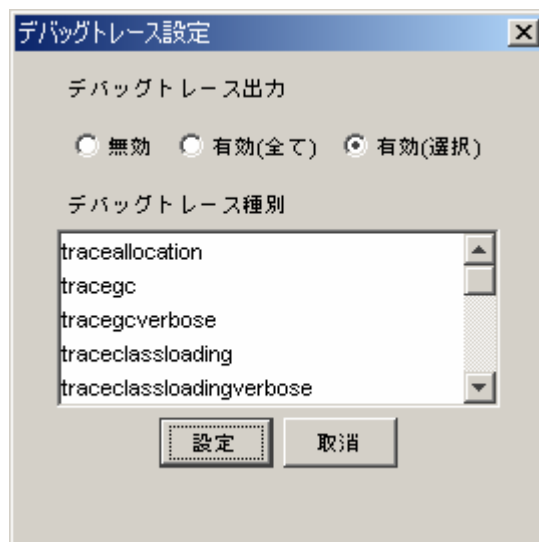


図 8-2 デバッグトレース設定画面

3. トレースしたいプロジェクトを読み込んで起動します。
（「4.4. プロジェクトの読み込み」を参照してください。）
4. トレース結果がコンソールに出力されます。
大量のトレース情報が表示されるので、解析の難しい複雑なデバッグ以外は「無効」にしてください。

第9章. iアプリ連携

「i アプリ連携」とは、i アプリから別の i アプリを起動することです。

i アプリ連携の起動方法には次の 2 通りがあります：

- 「ランチャモード」による起動
 「ランチャモード」とは、起動元アプリケーションからあらかじめ端末内にダウンロードされた起動先アプリケーションのエントリ ID を指定して起動するモードです。
 このモードは、起動先アプリケーションにパラメータを渡す事はできません。
 このモードは、`!Application#launch(LAUNCH_AS_LAUNCHER)` で起動できます。
- 「連携モード」による起動
 「連携モード」とは、起動元アプリケーションをダウンロードした同じホストからあらかじめ端末内にダウンロードされた起動先アプリケーションの ADF の URL を指定して起動するモードです。
 このモードは、起動先のアプリケーションにパラメータを渡す事ができます。
 このモードは、`!Application#launch(LAUNCH_IAPPLI)` で起動できます。

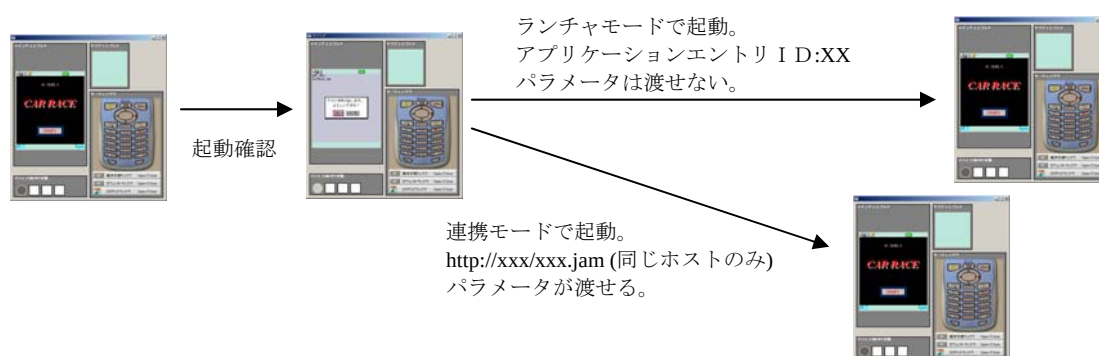


図 9-1 i アプリ連携の実行

i アプリ連携をするためには、以下の条件を満たしている必要があります：

- 起動元アプリケーションの ADF の項目で「LaunchApp」が「yes」に設定されていること。
- 起動元アプリケーションが即起動でないこと。
- 起動元アプリケーションが待ち受けアプリケーションの場合、非活性状態でないこと。
- ランチャモードの場合、起動先アプリケーションの ADF の項目で「LaunchByApp」が「deny」に設定されていないこと。
- 連携モードの場合、起動元アプリケーションの「トラステッド動作の設定」（「5.2 トラステッド動作の設定」を参照。）が 1 つでも"yes"に設定されている場合、起動先アプリケーションの ADF の項目で「TrustedAPIID」が設定されていること（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」を参照）。
- i アプリが「プロジェクト読み込み画面」で指定したプロジェクト格納先に作成されていること。必ずしもデフォルトの格納先（「C:\iDKDoJa5.0\apps」）に作成されている必要はありません。

第10章. ファイルの記述方法

本章では、変更可能な各種ファイルの記述方法を説明します。
各ファイルの場所は、以下のとおりです：

- ・ メール起動ファイル：
C:\iDKDoJa5.0\lib\mail*.mail
- ・ バウンダリデータファイル：
C:\iDKDoJa5.0\lib\mail\boundary.dat
- ・ メールアドレスファイル (※) :
C:\iDKDoJa5.0\lib\mail\myAddress.dat
- ・ ブラウザ起動HTMLファイル：
Web サーバの適切な場所
- ・ 着信履歴ファイル :
C:\iDKDoJa5.0\lib\phonehistory\phonehistory.lst

ここには、CallRecord#getLastRecord()で取得する
最新の着信履歴を指定します。
- ・ メールファイル (※) :
C:\iDKDoJa5.0\lib\mail\ 配下の
受信ボックス： received\[メッセージコード]\[受信日]_[通番].mail
送信ボックス： sent\[メッセージコード]\[送信日]_[通番].mail
未送信ボックス： unsent\[メッセージコード]\[送信日]_[通番].mail
- ・ 電話帳ファイル (※) :
C:\iDKDoJa5.0\lib\phonebook\[エントリ ID]_[名前].tel
- ・ 電話帳グループファイル (※) :
C:\iDKDoJa5.0\lib\group\[グループ ID]_[グループ名].grp
- ・ マルチメディア領域：
C:\iDKDoJa5.0\lib\multimedia
- ・ ブックマーク登録ファイル (※) :
C:\iDKDoJa5.0\lib\bookmark\[エントリ ID].bmk
- ・ スケジュール登録ファイル (※) :
C:\iDKDoJa5.0\lib\schedule\[エントリ ID].sdl
- ・ コード認識結果ファイル：
C:\iDKDoJa5.0\lib\multimedia\xxx.dat
(アプリ実行時にファイルダイアログで選択します。)
- ここには、CodeReader#read()で取得するコード認識の結果を指定します。
- ・ LED表示点灯パターン登録ファイル：
C:\iDKDoJa5.0\lib\phoneresource\info_[登録 ID].txt
- ・ ストレージデバイスファイルパス：
C:\iDKDoJa5.0\lib\storagedevice\xxx

- トルカファイル (※) :
C:\iDKDoJa5.0\lib\toruca\toruca[エントリ ID].dat
- ソーステンプレートファイル:
C:\iDKDoJa5.0\lib\template\xxx.txt
- 位置情報設定ファイル (※) :
C:\iDKDoJa5.0\lib\location\location_information\[エントリ ID].loc
- 位置情報画像起動設定ファイル (※) :
C:\iDKDoJa5.0\lib\location\location_information\[エントリ ID].img
- 測位データ設定ファイル (※) :
C:\iDKDoJa5.0\lib\location\survey_data\[通番].srv
C:\iDKDoJa5.0\lib\location\survey_data\survey_order (順序)
- スクリーンショットファイル:
C:\iDKDoJa5.0\lib\screenshot\scr_[通番].bmp
- イルミネーションデータファイル:
C:\iDKDoJa5.0\lib\illumination\illumination_[通番].twk
- デコメールテンプレートファイル (※) :
C:\iDKDoJa5.0\lib\decomailtemplate\Decom[エントリ ID].dmt
C:\iDKDoJa5.0\lib\decomailtemplate\entry.dat (タイトル一覧)
- コンパス設定ファイル (※) :
C:\iDKDoJa5.0\lib\location\compass.dat
- サウンド設定ファイル (※) :
C:\iDKDoJa5.0\lib\SoundConf.properties
- マイプロフィール設定ファイル (※) :
C:\iDKDoJa5.0\lib\myprofile.dat
- FeliCa アドホック通信設定ファイル:
C:\iDKDoJa5.0\lib\FelicaAdhoc.properties
- Bluetooth 設定ファイル (※) :
C:\iDKDoJa5.0\lib\bluetooth.properties
- DTV 連携起動パタメータファイル (※) :
C:\iDKDoJa5.0\lib\DTV.properties
- BML 連携起動パタメータファイル (※) :
C:\iDKDoJa5.0\lib\BML.properties
- 視聴/録画予約情報ファイル (※) :
C:\iDKDoJa5.0\lib\schedule\[エントリ ID].dtv
- 「アプリケーション名.properties」ファイル (※) :
C:\iDKDoJa5.0\apps\[アプリケーション名]\[アプリケーション名].properties
- 仮想 FeliCa データファイル (※) :
C:\iDKDoJa5.0\apps\[アプリケーション名]\[アプリケーション名].fc

- ・スクラッチパッドデータファイル (※) :
C:\iDKDoJa5.0\apps\[アプリケーション名]\sp\[アプリケーション名].sp
- ・ハイパーマルチデータファイル (※) :
C:\iDKDoJa5.0\lib\HyperMulti.properties
- ・加速度センサデータファイル (※) :
C:\iDKDoJa5.0\lib\location\acceleration\[通番].acc
C:\iDKDoJa5.0\lib\location\acceleration\acceleration_order
(順序)
- ・動作認識データファイル (※) :
C:\iDKDoJa5.0\lib\gesture\roll\[通番].ges
C:\iDKDoJa5.0\lib\gesture\roll\gesture_roll_order (順序)
C:\iDKDoJa5.0\lib\gesture\shake\[通番].ges
C:\iDKDoJa5.0\lib\gesture\shake\gesture_shake_order (順序)
- ・CSV入出力ファイル:
任意

※ このファイルは、iαppliTool (画面) から設定できます。

10.1. メール起動ファイルの記述方法

メール起動ファイル (*.mail) は、「4.16. メール起動」で使します。
サンプルで、C:\iDKDoJa5.0\lib\mail\0.mail 1.mail が用意されています。
メール起動ファイルの記述方法は、以下に示すとおりです。

1	From: xxx@docomo.ne.jp	送信者メールアドレス
2	To: zzz@docomo.ne.jp	受信者メールアドレス
3	Subject: i アプリ 起動 (IApplication)	Subject
4	Date: 2001/08/21 12:34	Date
5		
6	・・・メール本文・・・	本文
7		
8	--B:A	バウンダリ部
9	TEXT="iappli"	リンクを表示する際の文字列
10	ADF="http://*****/***.jam"	起動させる i アプリの ADF の URL
11	"A"="B"	i アプリに渡すパラメータ
12	"C"="D"	i アプリに渡すパラメータ

10.2. バウンダリデータファイル

バウンダリデータファイルは、メールに添付されたバウンダリ情報を格納するファイルです。
このファイルの内容は、Base64 でエンコードされた値でなければなりません。

10.3. メールアドレスファイル

メールアドレスファイルは、携帯電話に登録されたメールのアドレス情報を格納するファイルです。登録できるメールアドレスは1件のみです。

10.4. ブラウザ起動HTMLファイルの記述方法

ブラウザ起動HTMLファイルは、「4.1 7. ブラウザ起動」で使用します。
 サンプルで、C:\iDKDoJa5.0\lib\sample.html が用意されています。
 ブラウザ起動HTMLファイルの記述方法は、以下に示すとおりです。

1	<HTML>	
2	<BODY>	
3	<OBJECT declare id ="application.declaration"	OBJECT タグ
4	data="http://*****/***.jam"	起動させる i アプリの ADF の URL ※
5	type="application/x-jam">	
6	<PARAM NAME="A" VALUE="B">	
7	<PARAM NAME="C" VALUE="D">	
8	</OBJECT>	
9	Download	アプリを連携するリンク ista="#[OBJECT タグ]" href="[OBJECT タグに対応したHTML のURL]" (変更不要)
10	</BODY>	
11	</HTML>	

- ※ dataに "****.jam" のみを指定した場合、プロジェクトの読み込みで指定されたプロジェクト格納先に存在するアプリが起動されます。
- ※ ブラウザからメール、位置情報のパラメータを取得する事はできません。

10.5. 着信履歴ファイル

i アプリから取得できる着信履歴が格納されたファイルです。
着信履歴を変更したい場合は、このファイル内の履歴（4行で1履歴）を編集してください。
ただし、i アプリから取得できるのは、新着履歴のみですので、最初の1件（最初の4行）だけになります。

着信履歴ファイルの記述方法は、以下のとおりです。

1	2002/08/01 07:12	着信日時
2	true	着信の成功可否 "true" または "false"
3	1	電話帳エントリの I D
4	09012345678	電話番号または、"非通知設定"、"公衆電話" ※ CallRecord#getPhoneBookID()：この値を電話 帳に登録した同一エントリ I Dの電話番号と同じ 値にしてください。

※ 4行目の電話番号に "非通知設定" または、"公衆電話" の文字列を指定すると、i アプリからこの文字列がそのまま取得できます。

10.6. メールファイル

メールファイルは、送受信メールを格納するファイルです。
メールファイルの記述方法は、以下に示すとおりです。

1	From: xxx@docomo.ne.jp	送信者メールアドレス
2	To: xxx@docomo.ne.jp	受信者メールアドレス
3	Subject: i アプリ 起動 (IApplication)	Subject
4	Date: 2002/12/01 12:34	Date
5		
6	・・・メール本文・・・	本文
7		
8	--B:P	バウンダリ部
9	MessageCode="*****"	ADF のメッセージコード
10	data="*****"	バイナリデータ ※
11	--B:P--	

※ バイナリデータは、アプリケーション固有のデータです（「5.16. 送受信メールの編集」を参照）。

※ メールファイルの通番は、1～999で付与され、メッセージの I Dになります。

10.7. 電話帳ファイル

電話帳ファイルは、電話帳情報を格納するファイルです。
「電話帳」ファイルの記述方法は、以下に示すとおりです。

1	1	エントリ I D
2	姓	姓
3	名	名
4	ふりがな姓	ふりがな姓
5	ふりがな名	ふりがな名
6	0001112222	電話番号 1
7	0001112222	電話番号 2
8	0001112222	電話番号 3
9	mail@xxx.xxx.xx	メール 1
10	mail@xxx.xxx.xx	メール 2
11	mail@xxx.xxx.xx	メール 3
12	1	グループ I D
13	1104138322609	位置測位日時 (ミリ秒)
14	49	測位精度 (49 / 299 / 2147483647)
15	N00.00.00.000	緯度
16	E000.00.00.000	経度
17	0	測地系 0:WGS84 1:東京

10.8. 電話帳グループファイル

電話帳グループファイルは、電話帳グループ情報を格納するファイルです。
「電話帳グループ」ファイルの記述方法は、以下に示すとおりです。

1	1	エントリ I D
2	GROUP1	グループ名

10.9. マルチメディア領域

i アプリから `ImageStore#addEntry()` など、マルチメディア領域へ登録すると、以下のディレクトリ配下にファイルが作成されます。
また、マルチメディア領域から画像等を取得する場合、ファイル選択ダイアログが表示されるので、読み込みたいファイルを指定します。

マルチメディア領域：

C:\iDKDoJa5.0\lib\multimedia

この領域はそれぞれ以下のサブディレクトリに分かれていますので、i アプリで利用するファイル（画像等）を目的に応じて置いてください。

- ・アバターデータ : `avatar`
- ・i モーション : `imotion`
- ・マイピクチャ : `mypicture`
- ・サウンド : `sound`

なお、`imotion` および `mypicture` は、`DataBoxFolder API` にてフォルダ管理ができるようになっています。データフォルダは、「0000 フォルダ名」（0000:通番 4 桁）で作成されます。

「i α ppli Development Kit for DoJa-5.0(FOMA)」をインストールした時点で、マルチメディア領域には、いくつかのサンプルファイルが置いてあります。

10.10. ブックマーク登録ファイル

`BookMark#addEntry()` でブックマーク登録データが格納されたファイルです。
ブックマーク登録ファイルの内容は、以下のとおりです。

1	<code>http://xxx.xxx.xxx/xxx</code>	URL
2	<code>bookmark</code>	ブックマーク名

10.11. スケジュール登録ファイル

`Schedule#addEntry()` でスケジュール登録データが格納されたファイルです。
スケジュール登録ファイルの内容は、以下のとおりです。

1	1	スケジュールサポートタイプ
2	<code>schedule</code>	内容
3	年/月/週/日/時/分	スケジュール日時
4	1	アラームの有無 0:FALSE 1:TRUE

10.12. コード認識結果ファイル

バーコード認識のAPI (CodeReader)を利用する事により、本ファイルに指定したデータで、JANコードおよび、QRコードを認識する事ができます。

「iα ppli Development Kit for DoJa-5.0(FOMA)」は、実際にコードを読み取れる機能を持ちませんので、バーコード認識APIで読み取るデータを次の様なファイル形式でマルチメディア領域に作成しておきます（「10.9. マルチメディア領域」参照）。

sample1.dat

（拡張子が dat であればファイルの名前は何でも構いません。）

コード認識結果ファイル中のデータは、実際に読み込まれるデータを指定します。
読み込まれるデータは、それぞれ以下のとおりです。

1) JAN 8 コードの場合

XXXXXXXX	8桁の数字（ショートJAN）
----------	----------------

2) JAN 13 コードの場合

XXXXXXXXXXXX	13桁の数字
--------------	--------

3) QRコードの場合

[バイナリデータ]	本ファイルには、JPEG, GIF, MIDI(SMF), MFi のバイナリデータを直接指定します。 ▼ 実機で実際に認識するQRコードは以下の通りです。 (QRコード基本仕様 JIS X 0510 参照。) ・バージョン： 1 ～ 10 ・誤り訂正レベル： L、M、Q、H ・モデル： モデル2 ・対応モード： ① 数字：0～9(ASCII 値 0x30 ～ 0x39) ② 英数字：0～9(ASCII 値 0x30 ～ 0x39)、A～Zの26文字のアルファベット文字(ASCII 値 0x41～0x5A)、スペース、\$、%、*、+、-、,、/、:、の9個の記号(ASCII 値 0x20, 0x24, 0x25, 0x2A, 0x2B, 0x2D～0x2F, 0x3A) ③ 8-ビットバイト：JIS X 0201(ASCII 値 0x00～0xFF)に基づく文字 ④ 漢字：JIS X 0208に基づく Shift-JIS 漢字
-----------	---

以下のようなQRコードの解析結果によって、電話帳登録、Mail To、ブックマーク登録、iアプリ連携の4つの連携データを取得することができます。

エミュレータでは連携されませんが、実機では連携して動作します。

1) 電話帳登録の場合 (※)

MECARD:	電話帳登録を表す識別子
N: [姓], [名];	名前
SOUND: [フリガナ姓], [フリガナ名];	振り仮名
TEL: [12345678901234567890123456];	電話番号 (複数指定可)
TEL- <u>AV</u> : [12345678901234567890123456];	T V 発信用電話番号 (複数指定可) 《T V 電話機能付き移動機のみ》
EMAIL: [xxxxx@docomo.ne.jp];	メールアドレス (複数指定可)
NOTE: [メモ];	メモ 《対応機種のみ》
;	終端

2) Mail To (※)

MATMSG:	メール連携を表す識別子
TO: [xxxxx@docomo.ne.jp];	宛先
SUB: [題名];	表題
BODY: [あいうえお];	本文
;	終端

3) ブックマーク登録 (※)

MEBKM:	ブックマーク登録を表す識別子
TITLE: [NTTドコモ];	サイト名
URL: [http://www.nttdocomo.co.jp/];	URL
;	終端

4) iアプリ連携 (※)

LAPL:	i アプリ起動を表す識別子
ADFURL: [http\://xxx.xx.xx/xxx.jam];	起動する i アプリの ADF の取得先 URL 最低でも 255 バイトまで指定可。 (注) コードリーダー起動を実行する場合は、 http の後に \ を忘れず付与してください。
CMD: [起動コマンド];	起動コマンドへセットする文字列 最大 250 バイトまで指定可。
PARAM: [NAME, VALUE];	i アプリに渡すパラメータで名前と値をカンマで区切って指定する (複数指定可) 最大 255 バイト/件、16 件まで指定可。
;	終端

「iα ppli Development Kit for DoJa-5.0(FOMA)」は、KToolBar から「コードリーダー起動」を選択する事により、バーコードデータのパラメータを渡して i アプリを起動することができます (「4.18 コードリーダー起動」参照)。

※ 表記上改行していますが、実際には改行を入れないでください。

【記述例】

```
MECARD:N: ドコモ, 太郎;SOUND:ﾄﾞｺﾓ,ﾀﾔｳ;TEL:090XXXXXXXX;TEL:03XXXXXXXX;EMAIL:XXX@docomo.ne.jp;;
```


トルカファイルのフォーマットは以下のとおりです。

1	[ToruCa]	固定
2	Version=257	バージョン (2 バイト表記 上位バイト:メジャー 下位バイト:マイナー)
3	Type=0	トルカ種別 (0:SNIP / 1:CARD)
4	URL=http://xxx/xxx/xxx.rc	取得先URL
5	Data1=xxxxx	データ 1 (Base64 形式)
6	Data2=xxxxx	データ 2 (Base64 形式)
7	Data3=xxxxx	データ 3 (Base64 形式)
8	Cat=005D	カテゴリアイコン 2 バイト 16 進数表記
9	IP-ID=99999999	IP-ID
10	Sort-ID= X X	カナ ID
11	Dist=x	再配布識別子
12	Color-ID=0	色 ID 0 - 9
13	Expires=060101	有効期限 (Card のみ)
14	Body=[MIME-Version: 1.0 Content-Type: multipart/related;boundary="mimebounda rymarker" --mimeboundarymarker Content-Type: text/html; charset=Shift_JIS Content-Transfer-Encoding: 8bit <HTML> <HEAD> <TITLE>CarRace</TITLE> </HEAD> <BODY> <OBJECT declare id = "CarRace" data = "CarRace.j am" type = "application/x-jam"></OBJECT> <CENTER> SampleToruCa CarRa ce Start </CENTER> </BODY> </HTML> --mimeboundarymarker]	[]内に記述します (Card のみ) トルカ起動では、Body に指定 された OBJECT タグの i アプ リ (ADF) を起動します。 (右の例では、CarRace.jam) 注) 本フォーマット中では Body 部内の改行コードを "r\n" ("r'=0x0D, "n'=0x0A) から "t\v" ("t'= 0x09, "v'=0x0B) に置換して保存しておりま す。 そのため、直接ファイルを編 集する場合は、改行部分には" r\n"の代わりに"t\v"を記述し て下さい。 もしくは 5.2.1 トルカデー タ編集 の機能を用いて編集を 行うようにして下さい。(自 動的に置換・保存されま す。)

あからじめサンプルとして、SampleToruca.dat ファイルが置いてあります。

このフォーマットは、実際のトルカファイルとは異なります。

実際のトルカ仕様のバイナリデータを読み込ませたい場合は、HTTP 通信等でバイト配列として取得し、Toruca クラスのコンストラクタ `Toruca(byte[] data)` を利用して Toruca クラスを生成してください。

10.16. ソーステンプレートファイル

ソースファイルの新規作成時のテンプレートが格納されたファイルです。
このテンプレートファイルを変更することにより、「ソースファイルの作成」で作成されるソースファイルを変更することができます。
詳しくは、「4.2.1. ソースファイルの作成」を参照してください。

10.17. 位置情報ファイル

位置情報編集で登録された位置情報が格納されるファイルです。
格納される内容は、以下に示すとおりです。

エントリ ID	位置測位日時	測位レベル	緯度(※1)	経度(※1)	高度 (m)	測地系
位置履歴のエントリ ID	ミリ秒で測定した、測位時刻と 1970 年 1 月 1 日 午前 0 時 (GMT) との差	1 ～ 3 の数値	[-90 度, 90 度]	[-180 度, 180 度]	0 ～ の数値	0 (WGS84) 固定

各カラム間はカンマ区切りで指定します。

10.18. 位置情報画像登録ファイル

位置情報編集で画像に登録された位置情報が格納されるファイルです。
格納される内容は、以下に示すとおりです。

エントリ ID	位置測位日時	測位レベル	緯度(※1)	経度(※1)	高度 (m)	測地系
画像のエントリ ID	ミリ秒で測定した、測位時刻と 1970 年 1 月 1 日 午前 0 時 (GMT) との差	1 ～ 3 の数値	[-90 度, 90 度]	[-180 度, 180 度]	0 ～ の数値	0 (WGS84) 固定

各カラム間はカンマ区切りで指定します。

10.19. 測位データ設定ファイル

測位データ編集で登録された位置情報が格納されるファイルです。
格納される内容は、以下に示すとおりです。

測地系	緯度(※1)	経度(※1)	高度(m)	測位レベル	測位所要時間 (秒) (※2)	測地方式
0 (WGS84) 固定	[-90 度, 90 度]	[-180 度, 180 度]	-2147483648 ～ 2147483647 (-2147483648 は不明を示す)	1 ～ 3 の数値	0 ～ 2147483647	0 (GPS 測位方 式) 固定

各カラム間はカンマ区切りで指定します。

(※1) Location#getLatitudeString(PREFIX, UNIT)により定義されている DMS フォーマット
接頭辞は符号【(+) / - (+の時は表記無し)】・方位【N/S/W/E】両方使用できます。
dms(±ddd.mm.ss.sss) で表し、緯度は、北緯を+、南緯を-。経度は、東経を+、西
経を-。度を ddd、分を mm、秒を ss として小数点以下は 3 桁表記します。

測位レベルは、水平誤差 < 50m : レベル 3、水平誤差 < 300m : レベル 2、水平誤差
≥ 300m : レベル 3 です。

(※2) LocationProvider#getLocation() 実行時、測位データ取得までに要する時間です。
トラッキングモードの場合、この時間が [interval + timeout] (msec) を超える
と 測位不能 になります。

10.20. スクリーンショットファイル

スクリーンショットをファイルで保存する場合に作成されます。

「5.3.3. スクリーンショット保存の設定」で設定したパスにビットマップ形式のファイル
が蓄積されます。

10.21. イルミネーションデータファイル

i アプリ オプション API `IlluminationDisplay` を実行した場合に作成されます。
`IlluminationDisplay` で登録するとバイナリデータでファイルが作成されます。
登録されたファイルが必要ない場合は、手動で削除してください。

10.22. デコメールテンプレートファイル

デコメールテンプレートが保存されているファイルです。
ASCII 形式でデコメールテンプレートを格納します。
もし形式が正しくないと i アプリで例外が発生します。
デコメールテンプレートは MIME 形式で以下のように作成します。（例）

```
Decomail-Template
MIME-Version: 1.0
Content-Type: multipart/XXXXXXX; boundary="00000000"

--00000000
Content-Type: text/html; charset=Shift_JIS
Content-Transfer-Encoding: 8bit

<HTML><BODY>本文</BODY></HTML>

--00000000
Content-Type: image/xxxx; name="xxxxx.xxx"
Content-Transfer-Encoding: base64
Content-ID: <1>
インライン画像 1 の Base64 形式データ
:
--00000000--
```

} インライン画像がある場合

また、entry.dat に以下の形式でタイトルとデコメールテンプレートファイル名が保存されます。

デコメールテンプレートファイル名<タブ>タイトル

10.23. コンパス設定ファイル

電子コンパスの情報が格納されたファイルです。
DEGREE 単位で方位角を格納します。

10.24. サウンド設定ファイル

サウンド設定で設定されたデータが格納されるファイルです。
サウンド設定ファイルのフォーマットは以下のとおりです。

1	CH_1=SM_0	MF i のチャンネル 1 の代替音
:	:	:
16	CH_16=-1	MF i のチャンネル 1 6 の代替音
17	DEVICE=null	デバイス名
18	SOUNDLIB=001	サウンドライブラリ 001:汎用サウンドライブラリ 002:903i 向けサウンドライブラリ
19	MODE=0	再生モード 0:通常再生 1:ビジュアル

10.25. マイプロフィール設定ファイル

マイプロフィールが格納されたファイルです。
マイプロフィール設定ファイルのフォーマットは以下のとおりです。

1	true,NTT D o C o M o	参照種別(true/false), 名前 姓+名
2	true,NTT	参照種別(true/false), 名前の姓
3	true,D o C o M o	参照種別(true/false), 名前の名
4	true,エヌティティドコモ	参照種別(true/false), 名前 姓+名のカナ
5	true,エヌティティ	参照種別(true/false), 名前の姓のカナ
6	true,ドコモ	参照種別(true/false), 名前の名のカナ
7	true,00011112222	参照種別(true/false), 電話番号 1
8	true,11122223333	参照種別(true/false), 電話番号 2
9	true,message_app_usr1@docomo.ne.jp	参照種別(true/false), メールアドレス 1
10	true,message_app_usr2@docomo.ne.jp	参照種別(true/false), メールアドレス 2
11	true,100-6150	参照種別(true/false), 住所の郵便番号
12	true,東京都千代田区永田町 2-11-1 山王パークタワー	参照種別(true/false), 住所全部
13	true,東京都	参照種別(true/false), 住所の都道府県
14	true,千代田区永田町	参照種別(true/false), 住所の市町村
15	true,2-11-1	参照種別(true/false), 住所の番地
16	true,山王パークタワー	参照種別(true/false), 住所のマンション名
17	true,19920701	参照種別(true/false), 生年月日全部
18	true,1992	参照種別(true/false), 生年月日の年
19	true,07	参照種別(true/false), 生年月日の月
20	true,01	参照種別(true/false), 生年月日の日

10.26. 仮想 FeliCa アドホック通信設定ファイル

仮想 FeliCa のアドホック通信の通信制御情報が格納されたファイルです。
仮想 FeliCa アドホック通信設定ファイルのフォーマットは以下のとおりです。

1	SEND_SIZE=1024	1 回で送るデータ送信サイズ(バイト)
2	WAIT_TIME_SEND=100	1 回のデータ送信毎の待ち時間(ミリ秒)
3	RECV_SIZE=1024	1 回で受るデータ受信サイズ(バイト)
4	WAIT_TIME_RECV=100	1 回のデータ受信毎の待ち時間(ミリ秒)
5	PRE_WAIT_TIME_SEND=0	データ送信前の待ち時間 (ミリ秒)
6	PRE_WAIT_TIME_RECV=0	データ受信前の待ち時間 (ミリ秒)

仮想 FeliCa アドホック通信実行時に必要に応じて変更してください。

10.27. Bluetooth 設定ファイル

Bluetooth 通信の設定情報が格納されたファイルです。
Bluetooth 設定ファイルのフォーマットは以下のとおりです。

1	[OwnDevice]	自デバイスの情報
2	deviceType=516	自デバイスの端末属性 PC(260)/携帯電話(516)
3	deviceAddr=112233445566	自デバイスの Bluetooth アドレス
4	deviceName=DoJaBluetooth	自デバイスの名称
5		
6	[RegisteredDevice0]	登録デバイス(接続相手)の情報 1
7	deviceType=260	登録デバイス 1 の端末属性 PC(260)/携帯電話(516)
8	deviceAddr=112233445566	登録デバイス 1 の Bluetooth アドレス
9	deviceName=Terminal-1	登録デバイス 1 の名称
10		
:	[RegisteredDevice1]	登録デバイス(接続相手) を 10 件まで登録
:	:	
:	[RegisteredDevice9]	
:		
:	[ScanningDevice]	接続先の設定
:	host=localhost	Bluetooth 通信の相手先 IP アドレス

10.28. BML連携起動パラメータファイル

BML連携起動の設定をするファイルです。

BML連携起動パラメータファイルの記述方法は、以下のとおりです：

1	AFFILIATION_ID=1	系列識別 この値が連携先 i アプリの ADF : LaunchByBML に設定されている系列識別と一致する場合、BML ブラウザから連携先 i アプリを起動する
2	PARAMETER=xxxxx	i アプリに渡すパラメータ (省略可) 連携起動後 IApplication#getParameter("bml_param")で取得可能なパラメータを設定 省略した場合、上記メソッドはNULLを返す
3	JAM_NAME=test.jam	連携先 i アプリの jam ファイル名 BML ブラウザから連携起動する i アプリの jam ファイル名を設定 ※連携先 i アプリは現在のプロジェクト格納先に格納されている必要があります。

10.29. DTV連携起動パラメータファイル

DTV連携起動の設定をするファイルです。

DTV連携起動パラメータファイルの記述方法は、以下のとおりです：

1	#Wed Nov 15 10:00:00 JST 2006	ファイル更新時刻
2	service_id=1	サービス識別 (省略可) 連携起動後 IApplication#getParameter("service_id")で取得可能なパラメータを設定 省略した場合、上記メソッドはNULLを返す
3	frequency=13	物理チャンネル (省略可)
4	affiliation_id=1	系列識別 (省略可)
5	ADF= test.jam	連携先の番組表 i アプリの jam ファイル名 DTV から連携起動する i アプリの jam ファイル名を設定 ※連携先 i アプリは現在のプロジェクト格納先に格納されている必要があります。

DTV からの連携による i アプリの起動・レジューム後に DTVParameter#getLastParameter() を実行した場合、このファイルに指定した最新のパラメータが取得されます。

ファイルが存在しなかった場合、DTVParameter#getLastParameter() は NULL を返します。

10.30. 視聴/録画予約情報ファイル

DTVSchedule#addEntry() で視聴/録画予約スケジュール登録データが格納されたファイルです。
視聴/録画予約情報ファイルの内容は、以下のとおりです:

1	1	予約タイプ (0: 視聴予約、1: 録画予約)
2	X X X X X X X X X X	チャンネル名
3	1	サービス識別 (0 ~ 65535)
4	13	物理チャンネル番号 (13 ~ 62)
5	1	系列識別 (0 ~ 11)
6	年/月/週/日/時/分	番組開始時刻 (月は0 ~) これらのフィールドのうち実際に出力されるのは ScheduleDate に設定されたフィールドのみであ り、他は空となる 月フィールドは1~12ではなく、1月を0とする0 ~11の範囲の値で表現される 週フィールドは、日曜日を1とする1~7の範囲の 値で表現される
7	年/月/週/日/時/分	番組終了時刻 (月は0 ~) 予約タイプが視聴予約の場合は省略される これらのフィールドのうち実際に出力されるのは ScheduleDate に設定されたフィールドのみであ り、他は空となる 月フィールドは1~12ではなく、1月を0とする0 ~11の範囲の値で表現される 週フィールドは、日曜日を1とする1~7の範囲の 値で表現される
8	X X X X X X X X X X	番組名
9	1	繰り返し設定 (ScheduleDate のタイプ)

10.31. 「アプリケーション名.properties」ファイル

端末固有の設定をこのファイルに記述します。

「アプリケーション名.properties」ファイルの記述方法は、以下のとおりです。

1	appMoveInfo = false	アプリケーションがメディア等により移動機外部から移されたものかどうかを指定 (true / false)
2	phoneBookEntry = 10	電話帳エントリの最大保存件数
3	xDcmMove = 0000	メディアリソースのHTTPヘッダ "X-Dcm-Move" で、0000 か 0001 を指定
4	linkHTTPHeader = false	拡張HTTPヘッダ (x-Dcm-Move) の関連付け (true / false)
5	cameraFrameRate = 15	カメラのフレームレート
6	moveFromOtherTerm = false	移動先と移動元の移動機が異なるかどうかを指定 (true / false)
7	adfUrl = http://*****	起動させる i アプリの ADF の URL
8	receivedMailMax = 200	受信メールの最大保存件数
9	sentMailMax = 50	送信/未送信メールの最大保存件数
10	accelerationDataMax = 20	加速度センサデータの最大保存件数
11	bookmarkEntry = 10	ブックマークの最大保存件数
12	launchType = 0	i アプリ読み込み時の起動タイプ
13	groupEntry = 10	グループの最大保存件数
14	dtvScheduleMax = 50	視聴/録画予約の最大保存件数
15	remoteParameter =	リモート制御の受信パラメータ
16	remotePushedTime =	リモート制御の受信時間
17	sound3dResource = 4	3D サウンドの制御を行うためのリソース数
18	cameraDevice = 1	カメラデバイス数
19	launchTime =	i アプリのスケジューラ起動時刻 n@YYYY/MM/DD/hh/mm/ss n:設定番号 1 ~ ; で区切って複数指定
20	launchTimeMax = 4	スケジューラ起動時刻の最大数 (4 固定)
21	fontBaselinePos =	フォントベースライン設定
22	stubSourceName = XXX	i アプリライブラリのスタブファイル名
23	componentVendor = XXX	i アプリライブラリの会社名
24	componentTitle = XXX	i アプリライブラリの名前
25	componentName = XXX	i アプリライブラリの Jar ファイル名
26	componentAuthor = XXX	i アプリライブラリの作者
27	componentVersion = XXX	i アプリライブラリのバージョン
28	componentCreate = yes	i アプリライブラリの有効/無効 (true / false)

10.32. 仮想 FeliCa データファイル

仮想 FeliCa データファイルは、FeliCa チップを仮想的に実現したもので、このチップデータをファイルで管理しています。仮想 FeliCa データファイルは、Felica#open()を実行すると自動的に作成されます。

本エミュレータで扱える仮想 FeliCa データファイルは、SystemDefineBlock と AreaDefineBlock を先頭に 3 つのフリーエリアに分かれています。

仮想 FeliCa データファイルのフォーマットは以下のとおりです。

領域	属性	ブロック数	備考
フリー領域	SystemDefineBlock	1	先頭 2 バイトは、SystemCode
	AreaDefineBlock	1	先頭 4 バイトは、Area0000
	AreaDefineBlock	1	Data Block を除き、 先頭 4 バイトは、 AreaCode/ServiceCode
	PIN Service/PIN DATA	2	
	Service Block 1	1	
	PIN Service/PIN DATA	2	
	CPID Data Block	1	
	Service Block 2	1	
	PIN Service Block	1	
	PIN Data Block	1	
	Data Block 0	1	
	Data Block 1	1	
	Data Block 2	1	
	Data Block 3	1	
	AreaDefineBlock	1	Data Block を除き、 先頭 4 バイトは、 AreaCode/ServiceCode
	PIN Service/PIN DATA	2	
	Service Block 1	1	
	PIN Service/PIN DATA	2	
	CPID Data Block	1	
	Service Block 2	1	
	PIN Service Block	1	
	PIN Data Block	1	
	Data Block 0	1	
	Data Block 1	1	
	Data Block 2	1	
	Data Block 3	1	
	AreaDefineBlock	1	Data Block を除き、 先頭 4 バイトは、 AreaCode/ServiceCode
	PIN Service/PIN DATA	2	
	Service Block 1	1	
	PIN Service/PIN DATA	2	
	CPID Data Block	1	
	Service Block 2	1	
	PIN Service Block	1	
	PIN Data Block	1	
	Data Block 0	1	
	Data Block 1	1	
	Data Block 2	1	
	Data Block 3	1	

1 ブロック = 1 6 バイト

SystemDefineBlock = システム定義ブロック
 AreaDefineBlock = エリア定義ブロック
 Service Block 1 = CPID が格納されているサービスのサービス定義ブロック
 CPID Data Block = CPID が格納されているユーザブロック
 Service Block 2 = CP がデータを格納するサービスのサービス定義ブロック
 Data Block 0～3 = CP がデータを格納するサービスのユーザブロック
 PIN Service Block = PIN サービスのサービス定義ブロック
 PIN Data Block = PIN が格納されているユーザブロック

仮想FeliCaデータファイルの内容を編集するには、仮想FeliCaデータファイルビューワを使用します（「6.3. 仮想FeliCaデータファイルビューワ」を参照）。

10.33. スクラッチパッドデータファイル

スクラッチパッドのデータを格納しておくファイルです。
このファイルは、i アプリからスクラッチパッドに書き込む事で作成されます。

スクラッチパッドの編集方法については、「6.1. スクラッチパッド管理ツール」を参照してください。

10.34. ハイパーマルチデータファイル

ハイパーマルチ機能を設定したデータが格納されるファイルです。

「ハイパーマルチデータファイル」に格納される内容は、以下に示すとおりです。

1	Enable = true	ハイパーマルチ有効 : true / false
2	Mode = 1	セバレートモード : 1, 2, 3

10.35. 加速度センサデータファイル

加速度センサ機能で取得するデータが格納されるファイルです。

「加速度センサデータファイル」に格納される内容は、以下に示すとおりです。

1	100	X 軸方向
2	200	Y 軸方向
3	300	Z 軸方向
4	60	ロール角
5	80	ピッチ角
6	90	向き (0, 90, 180, 270, -1)

各カラム間はカンマ区切りで指定します。

10.36. 動作認識データファイル

動作認識機能で利用するデータが格納されるファイルです。

「動作認識データファイル」に格納される内容は、以下に示すとおりです。

Roll 動作

X 方向位置	Y 方向位置	品質
整数	整数	0 ～ 3

各カラム間はカンマ区切りで指定します。

Shake 動作

振動量
0 ～ 511

10.37. CSV入出力ファイル

測位データ編集、加速度センサデータ編集、動作認識データ編集において、インポートもしくはエクスポートするCSV形式のファイルです。

■測位データ

測位データのCSVファイルは、以下のデータです。

緯度(度)	経度(度)	高度(m)	測位レベル	測位所要時間(秒)	測地系	測地方式
[-90～90]	[-180～180) または [-180～179]	-2147483648 ～ 2147483647	1 ～ 3	0 ～ 2147483647	0(WGS84) 固定	0(GPS 測位) 固定

■加速度センサデータ

加速度センサデータのCSVファイルは、以下のデータです。

X 軸方向	Y 軸方向	Z 軸方向	ロール角	ピッチ角	向き
数値	数値	数値	-180～179	-180～179	-1, 0, 90, 180, 270

■動作認識データ

動作認識データのCSVファイルは、以下のデータです。

・Roll 動作

X 方向位置	Y 方向位置	品質
数値	数値	0 ～ 3

・Shake 動作

振動量
0 ～ 511

10.38. ファイルの記述具体例

「http://xxx.yyy.zzz」の「CarRace.jam」をメール、ブラウザ、コードリーダーから起動する場合:

- メール起動
 - 起動したい i アプリ (CarRace) の ADF の「LaunchByMail」に「any」を設定します。
 - メール起動ファイルに「C:\iDKDoJa5.0\lib\mail\0.mail」を用います。
 - 「0.mail」ファイルの ADF には、
「ADF="http://xxx.yyy.zzz/CarRace.jam"」と記述します。
 - メール起動で、[Sample1](#) を起動します。
- ブラウザ起動
 - 起動したい i アプリ (CarRace) の ADF の「LaunchByBrowser」に「any」を設定します。
 - ブラウザ起動 HTML ファイルに「C:\iDKDoJa5.0\lib\sample.html」を用います。
 - 「sample.html」の OBJECT タグの data には、
「data="http://xxx.yyy.zzz/CarRace.jam"」と記述して、
<OBJECT>タグにダウンロードして起動する i アプリの ADF を記述します。
(プロジェクト読み込み先に CarRace が存在する場合は、
「data="CarRace.jam"」でも起動可能です。)
 - ブラウザ起動で、URL に「file:c:/iDKDoJa5.0/lib/sample.html」を指定して起動します。
- コードリーダー起動
 - 起動したい i アプリ (CarRace) の ADF の「AllowPushBy」に
「Code:command」(command は 1 0.1 2. コード認識結果ファイルの CMD: の値)を設定します。
 - コード認識結果ファイルに
「C:\iDKDoJa5.0\lib\multimedia\sample1.dat」を用います。
 - 「sample1.dat」ファイルの起動する i アプリ (ADFURL:) には、
「ADFURL:http://xxx.yyy.zzz\CarRace.jam」と記述します。
 - コードリーダー起動で、sample1.dat を起動します。
- トルカ起動
 - 起動したい i アプリ (CarRace) の ADF の「LaunchByToruca」に「any」を設定します。
 - トルカファイルに「C:\iDKDoJa5.0\lib\toruca\SampleToruca.dat」を用います。
 - 「SampleToruca.dat」ファイルの Body 部の Object タグに、
「data="CarRace.jam"」と記述します。
 - トルカ起動で、SampleToruca.dat を選択して起動します。

2. 図 1 1-1 の「コンテンツ」タブを選択します（図 1 1-2）。

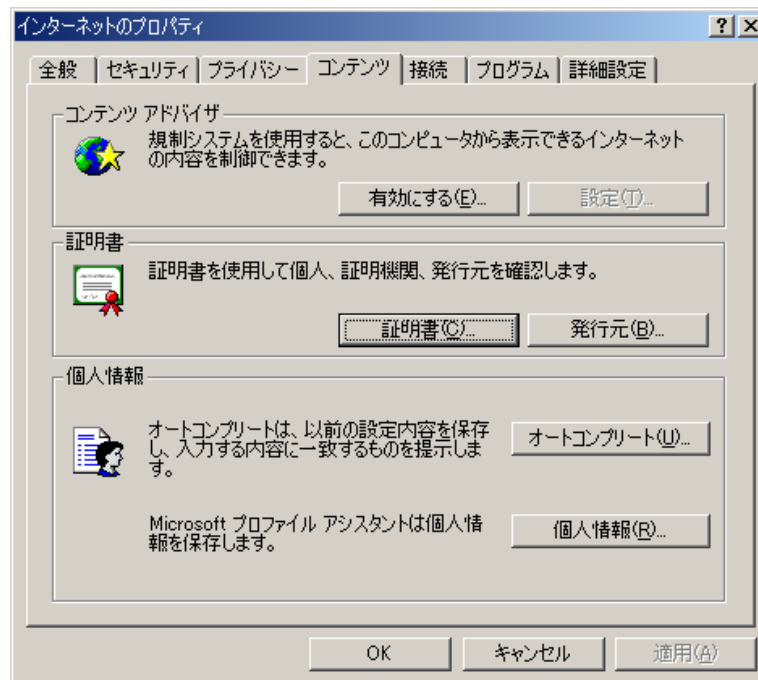


図 1 1-2 インターネットのプロパティ（コンテンツ）

3. 図 1 1-2 の「証明書 (C) ...」 ボタンを押します（図 1 1-3）。

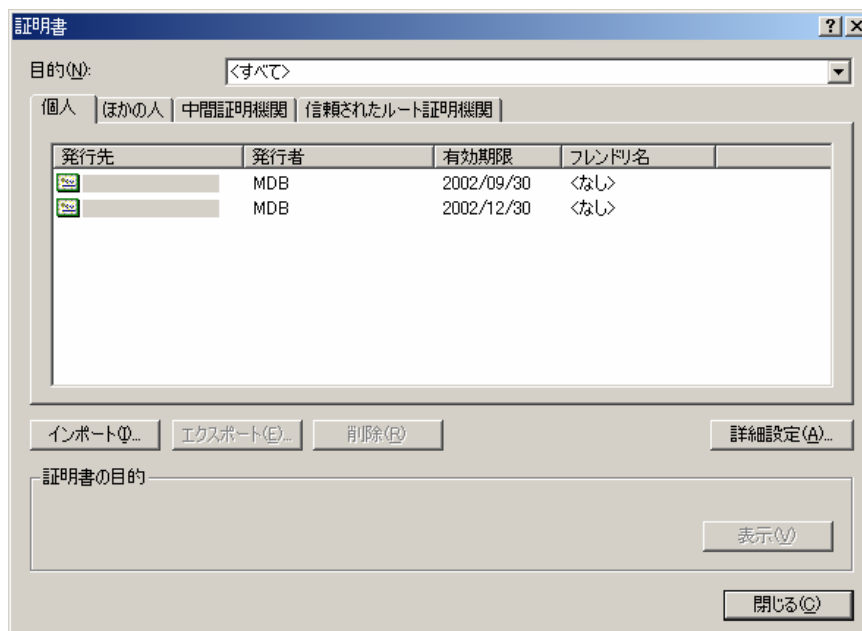


図 1 1-3 証明書画面

4. 図 1 1-3 の証明書画面で証明書の管理を行います。使用方法については、OSのドキュメントを参照してください。

第12章. OBEX の利用

「iα ppli Development Kit for DoJa-5.0(FOMA)」では、TCP/IP もしくは赤外線(IrDA)を用いて OBEX 通信することができます。

TCP/IP を用いて OBEX 通信を行う場合、「iα ppliTool」の実行ファイルが置かれているディレクトリ（「C:\iDKDoJa5.0\bin」に置かれる）にある libobex_tcp.dll を libobex.dll に名前を変更してください。通信先のホストの設定は、「iα ppliTool」のメニューから「設定」→「ネットワークの設定...」の選択で表示される画面の「OBEX 設定」で行います。何も設定しない場合には、127.0.0.1（ローカルホスト）に通信します。使用される TCP のポート番号は 650 番です。ポート番号を変更することはできません。

Sun Java Studio Enterprise 8 もしくは、NetBeans IDE 4.1 での設定は「1 4.4.1. デバイスの設定」を参照してください。

赤外線を用いて OBEX 通信を行う場合、「iα ppliTool」の実行ファイルが置かれているディレクトリにある libobex_irda.dll を libobex.dll に名前を変更してください。赤外線を利用する場合は、ホストの設定の必要はありません。

なお、インストール時の libobex.dll は libobex_tcp.dll であり、TCP/IP での通信となります。

第13章. 再配布不可識別子

i アプリのダウンロードやメールの添付などで取得したメディアデータ（画像、動画や音楽）には、i モード携帯電話より外に出させないようにするため、再配布不可識別子を付与することができます。

エミュレータでは、GIF(89a、87a)、JPEG (EXIF、JFIF)、MFi、SMF、MP4 のメディアデータに対して再配布不可識別子のチェックをしています。

読み込んだメディアデータが再配布不可であるかどうかは、MediaResource#isRedistributable() を実行することで確認することができます。

エミュレータでは、selectEntry()、getEntry()を実行する際には再配布不可識別子のチェックは行いません。

また、マルチメディア領域へのデータの保存時にメディアデータ、トルカ、デコメールテンプレートへの再配布不可識別子の設定は行いません。

第14章. Sun Java Studio Enterprise / NetBeans IDE の利用

この章では、Sun Java Studio Enterprise、NetBeans IDE 4.1 と「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」の拡張モジュールについて説明します。
(最新の NetBeans IDE 5.0 でも利用可能ですが動作が保証されません。)

14.1. Sun Java Studio Enterprise および NetBeans IDE について

Sun Java Studio Enterprise 8 もしくは、NetBeans IDE 4.1 に「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」をインストールする事で、Sun Java Studio Enterprise 8 もしくは、NetBeans IDE 4.1 からエミュレータを直接起動することができます。「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」は、Sun Java Studio Enterprise 8 と NetBeans IDE 4.1 で共通に使える i アプリを作成するための拡張モジュールです。

14.2. 「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」のインストール

この節では、「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」のインストール方法について説明します。

14.2.1. インストール

「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」をインストールするには、まず、「2.2. インストール手順」のセットアップタイプの選択画面（図 2-6）で、「カスタム」を選択し、機能の選択画面で、「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール」をチェックします（図 14-1）。

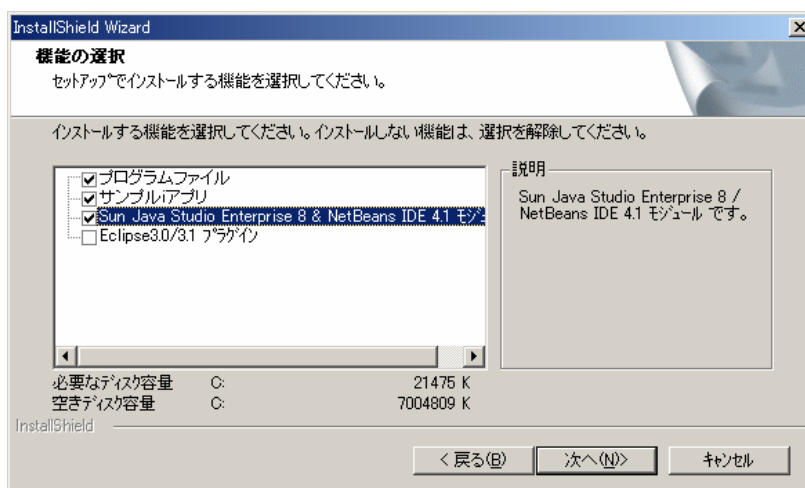


図 14-1 Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール のインストール

注意：・ **本モジュールはデバッガには非対応です。**
そのため i アプリにはデバッガ情報は含まれません。
・ Sun Java Studio Enterprise は、JDK5.0 以降でのみ動作可能です。

14.2.2. IDE にモジュールを組み込む

「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」は、次の場所に置かれています。

➤ C:\iDKDoJa5.0\plugin\org.netbeans.modules.iappli.nbm

このモジュールは、Sun Java Studio Enterprise もしくは、NetBeans IDE の機能「アップデートセンター」で組み込む事によって使えるようになります。そのため、以下の手順を踏まないと IDE で i アプリを作成できませんので、必ず実施してください。

もし、旧モジュール(DoJa-4.1)がインストールされている場合は、これをアンインストールしてください。この場合、Sun Java Studio Enterprise もしくは、NetBeans IDE を再起動してからインストールください。

IDE に「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」を組み込むには、次の手順に従います。

1. Sun Java Studio Enterprise もしくは、NetBeans IDE を起動します。
2. 「ツール」→「アップデートセンター」を選択します（図 14-2）。

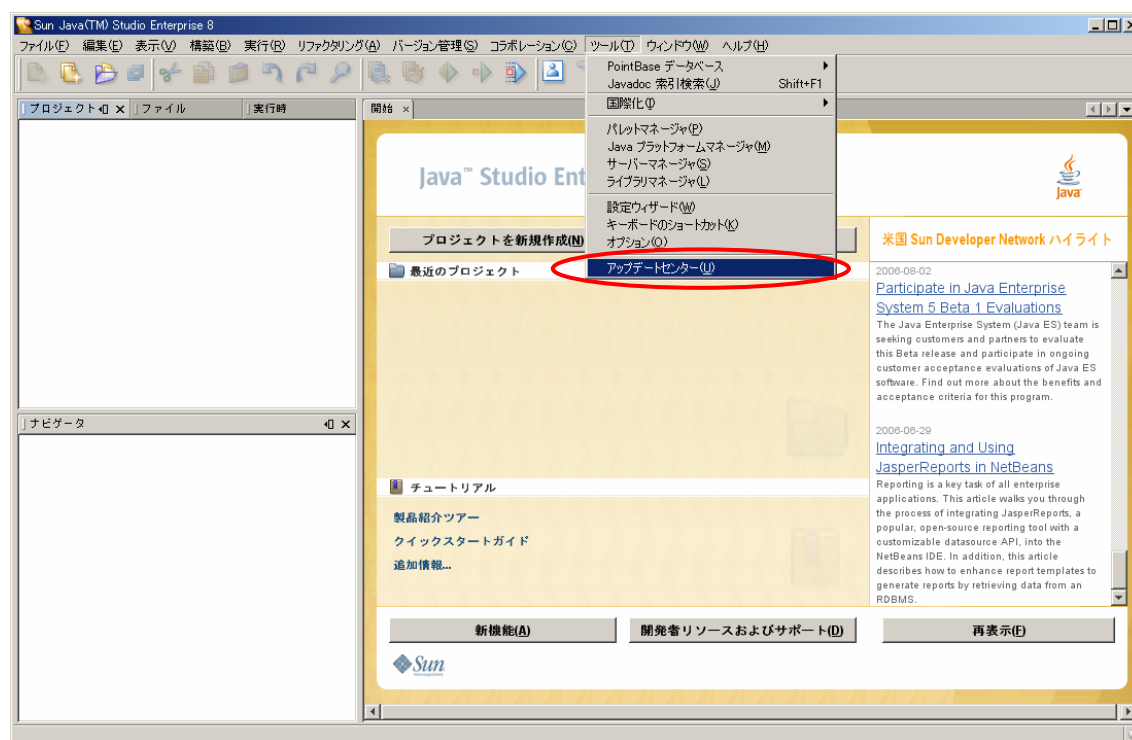


図 14-2 アップデートセンターの実行

3. 「モジュールの場所を選択」で、「手動でダウンロードしたモジュールをインストール」にチェックを付けて、「次へ」を押します（図 1 4-3）。

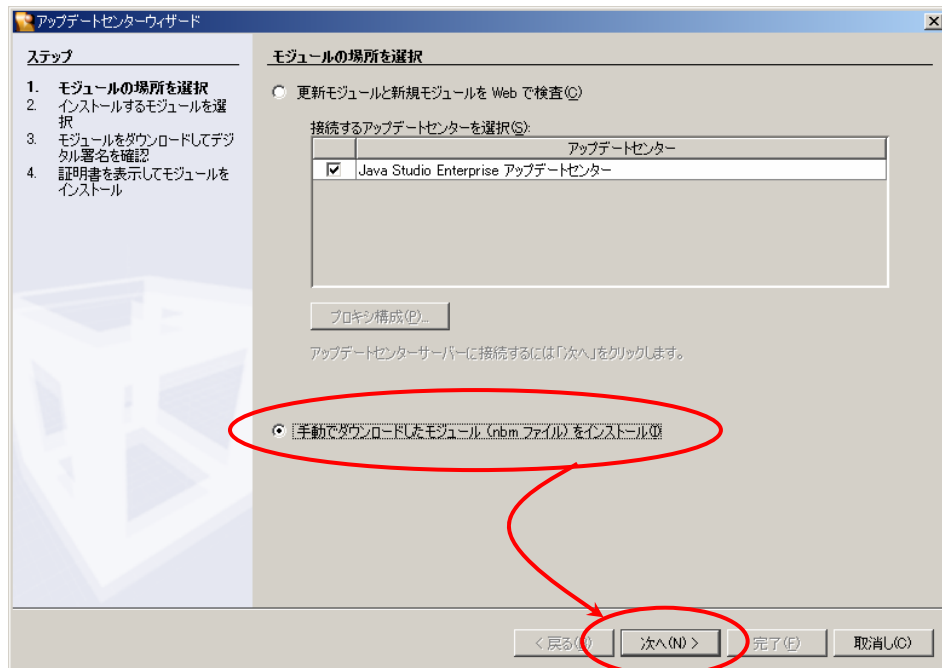


図 1 4-3 モジュールの場所を選択

4. 「インストールするモジュールを選択」で、「追加」ボタンを押して、「C:\jdk\DoJa5.0\plugin\org.netbeans.modules.iappli.nbm」を追加します（図 1 4-4）。

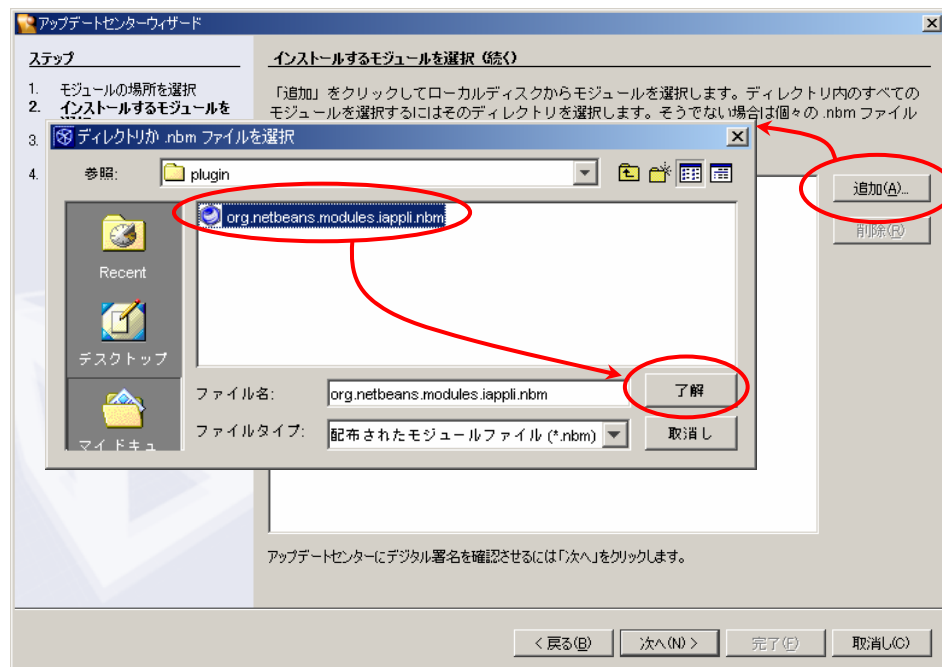


図 1 4-4 モジュールの選択

5. 「次へ」を押すと、モジュールの確認画面が表示されるので、「インストールを含む」の一覧に **DoJa-5.0** が含まれていることを確認して、「次へ」を押します（図 1 4-5）。

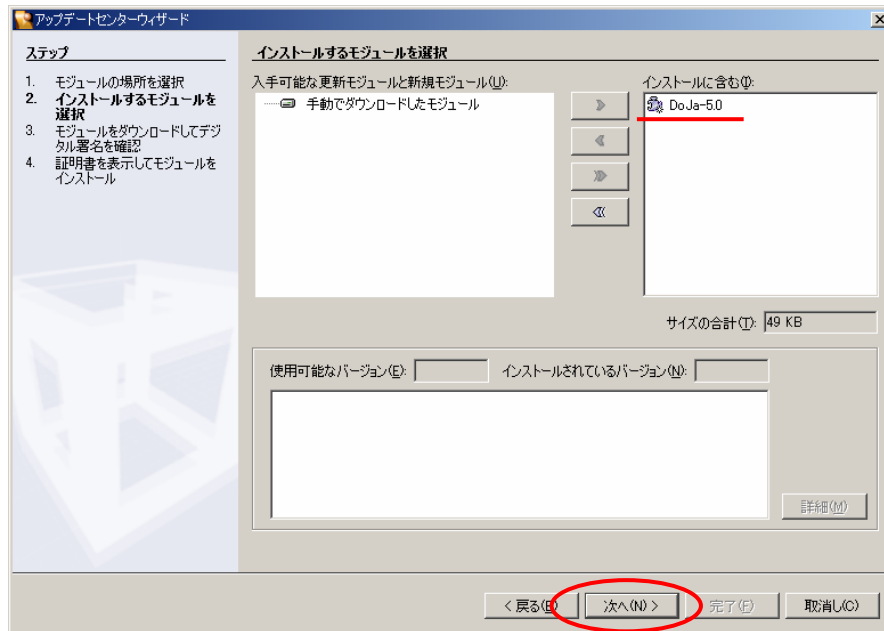


図 1 4-5 モジュールの確認

6. デジタル署名の確認が実行されるので、完了したら「次へ」を押します（図 1 4-6）。

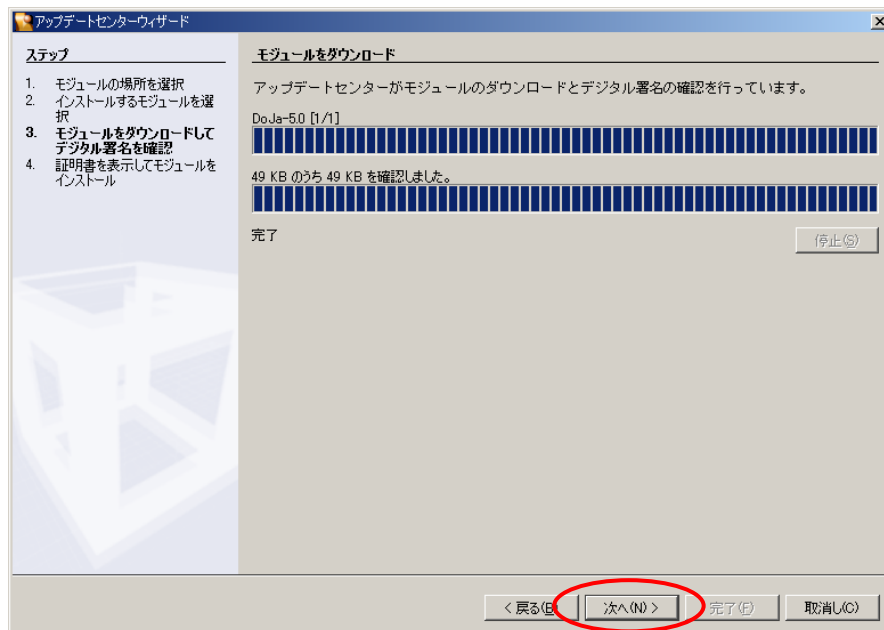


図 1 4-6 モジュールのデジタル署名

7. インストールするモジュールの一覧から「DoJa-5.0」のモジュールにチェックを付けて、「完了」を押します。
この時、署名がない旨の確認が表示されますので「はい」を押してください（図 1 4-7）。

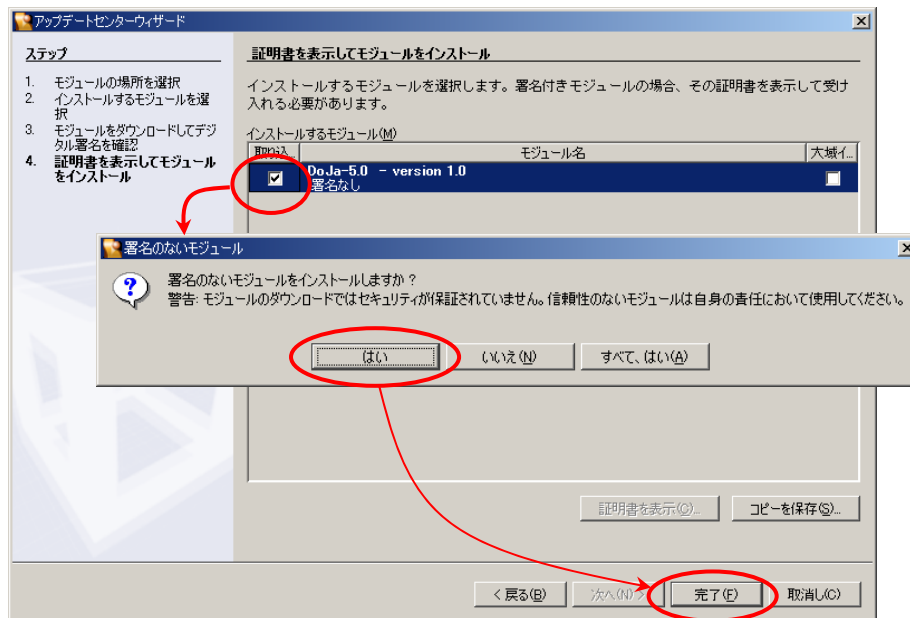
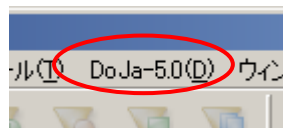


図 1 4-7 モジュールのインストール

8. しばらくすると、Sun Java Studio Enterprise もしくは、NetBeans IDE に DoJa-5.0 メニューが表示されます（多少時間がかかる場合があります）。



9. エミュレータを「C:\iDKDoJa5.0」以外にインストールした場合は、「DoJa-5.0」→「環境設定」を選択して、「DoJa-5.0 のインストール先」を正しいパスに変更してください（図 1 4-8）。

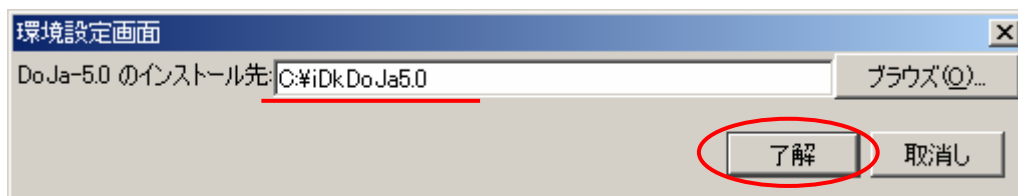


図 1 4-8 エミュレータのインストール先 設定

以上で、「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for i α ppli」の組み込みは完了です。

14.3. 新規アプリケーションの作成

この節では、新しいアプリケーションの作成方法について説明します。

14.3.1. プロジェクトの作成

アプリケーションのソースファイルと属性ファイルを収容するパッケージを作成します。
パッケージを作成するには、次の手順に従います。

1. 「ファイル」→「新規プロジェクト」を選択します（図 1 4-9）。

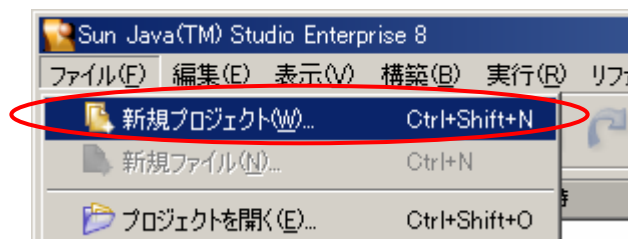


図 1 4-9 新規プロジェクトメニュー

2. 「DoJa」 - 「DoJa-5.0プロジェクト」を選択して、「次へ」を押します（図 1 4-10）。

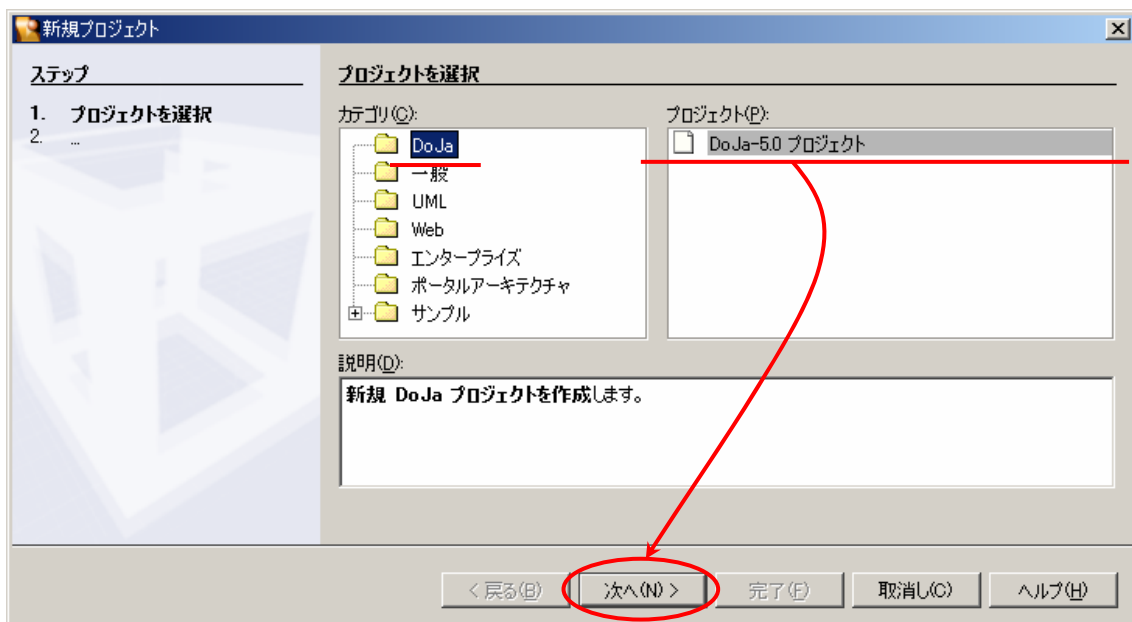


図 1 4-10 プロジェクトの種類

3. 「プロジェクト名」を入力して、「完了」を押します（図 1 4-1 1）。
例えば、プロジェクト名に「hello」と入力します。

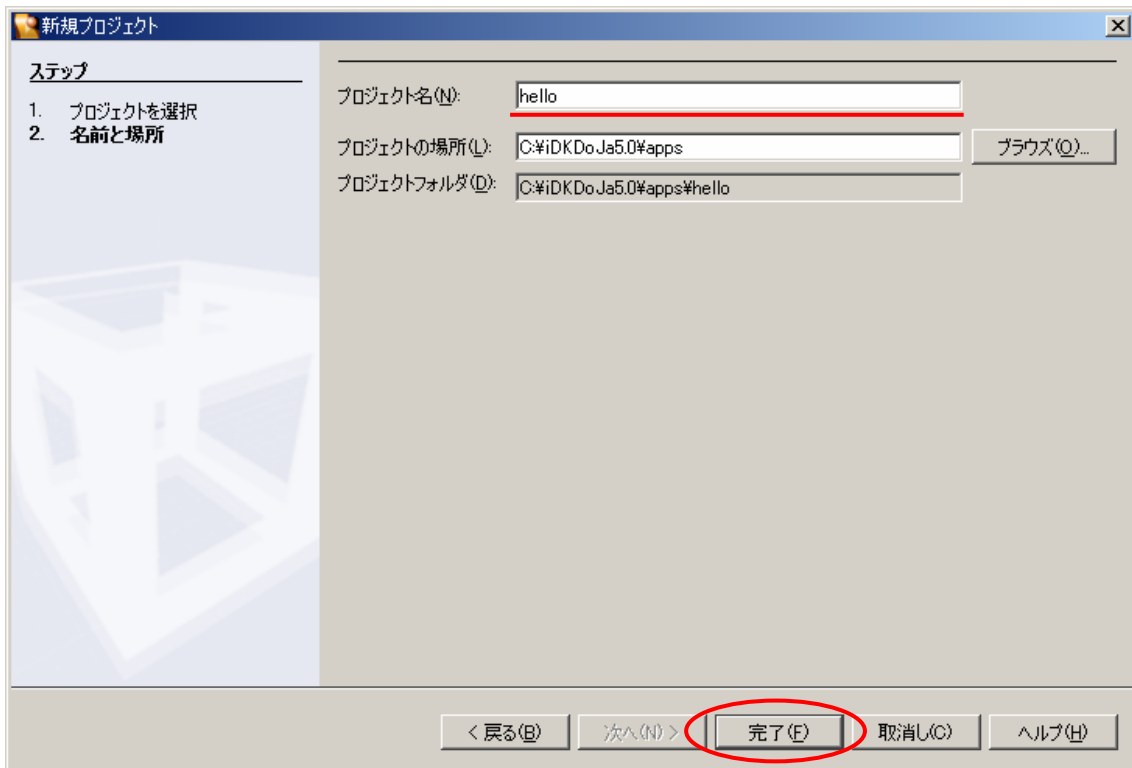


図 1 4-1 1 プロジェクト名とプロジェクトの場所

4. 作成したプロジェクトが表示されます（図 1 4-1 2）。

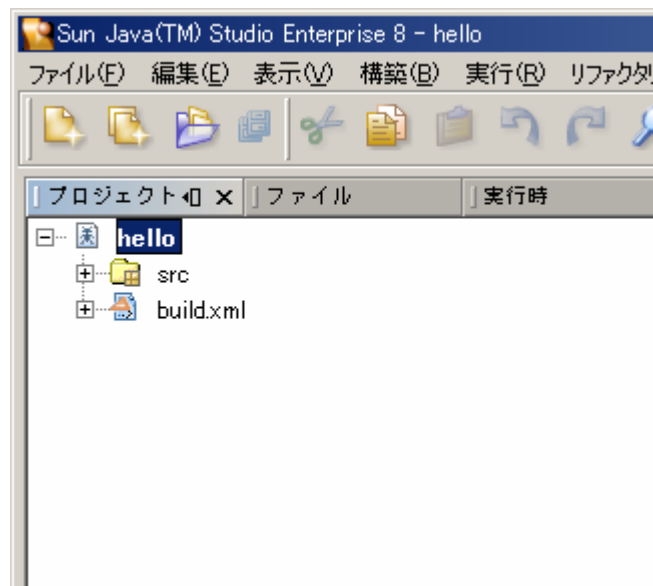


図 1 4-1 2 プロジェクトウィンドウ

14.3.2. ソースファイルの作成

アプリケーションのソースファイルを作成するには、次の手順に従います。

1. プロジェクトウィンドウの「src」を右クリックして、「新規」→「Java クラス」を選択します。

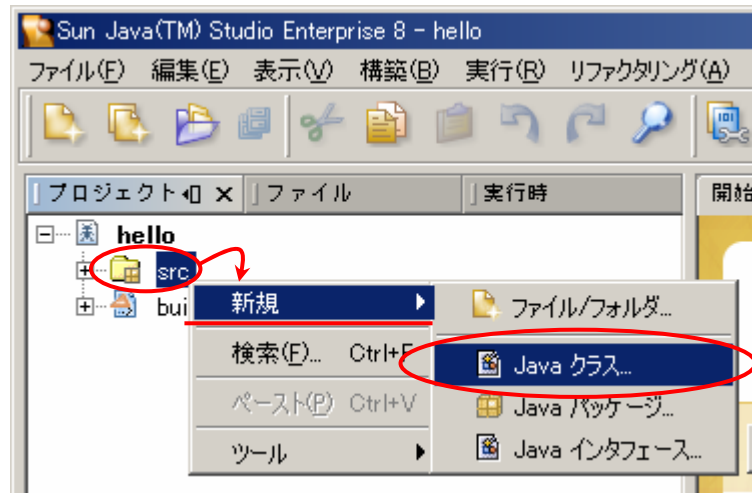


図 1 4-1 3 Java ソースファイルの作成

2. クラス名とパッケージ名を入力して、「完了」を押します。
例えば、クラス名に「HelloWorld」、パッケージ名に「hello」と入力します。

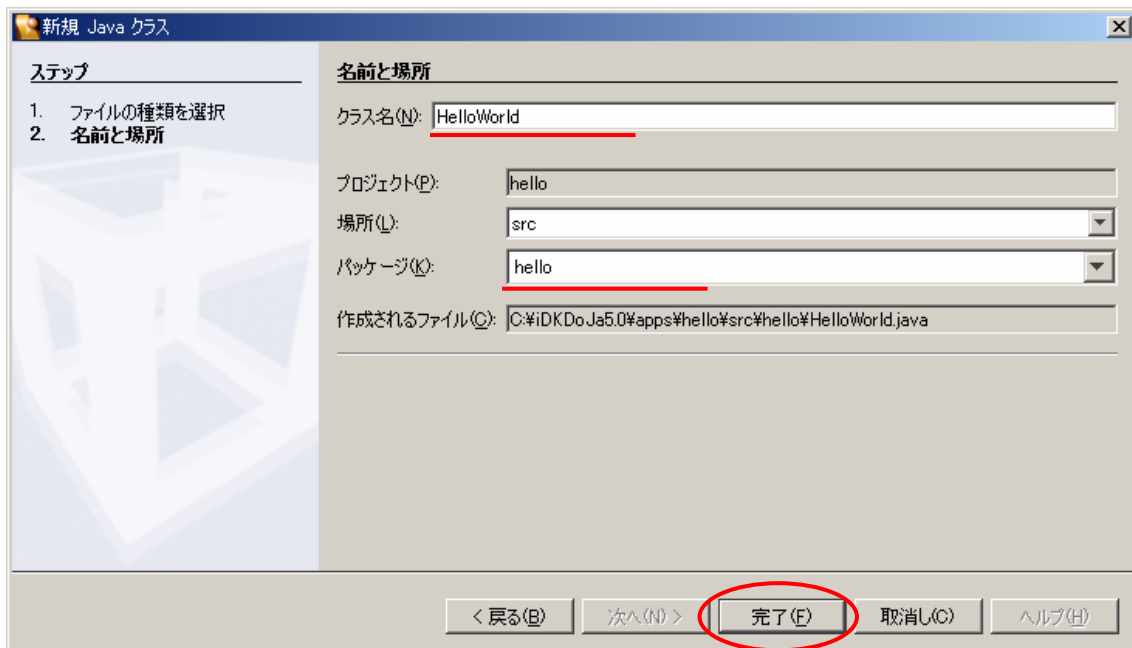


図 1 4-1 4 クラス名の設定

3. プロジェクトに空のクラスのソースファイルが追加されます。
 たとえば、ソースファイル中で、
 package hello; の後に
import com.nttdocomo.ui.IApplication;
 public class HelloWorld に続けて、
extends IApplication を追加し、
 public class HelloWorld extends IApplication { の後に
public void start() {
 System.out.println("Hello World");
}
 を追加します。
 また、コンストラクタ **public HelloWorld() { }** は不要なので削除 します。

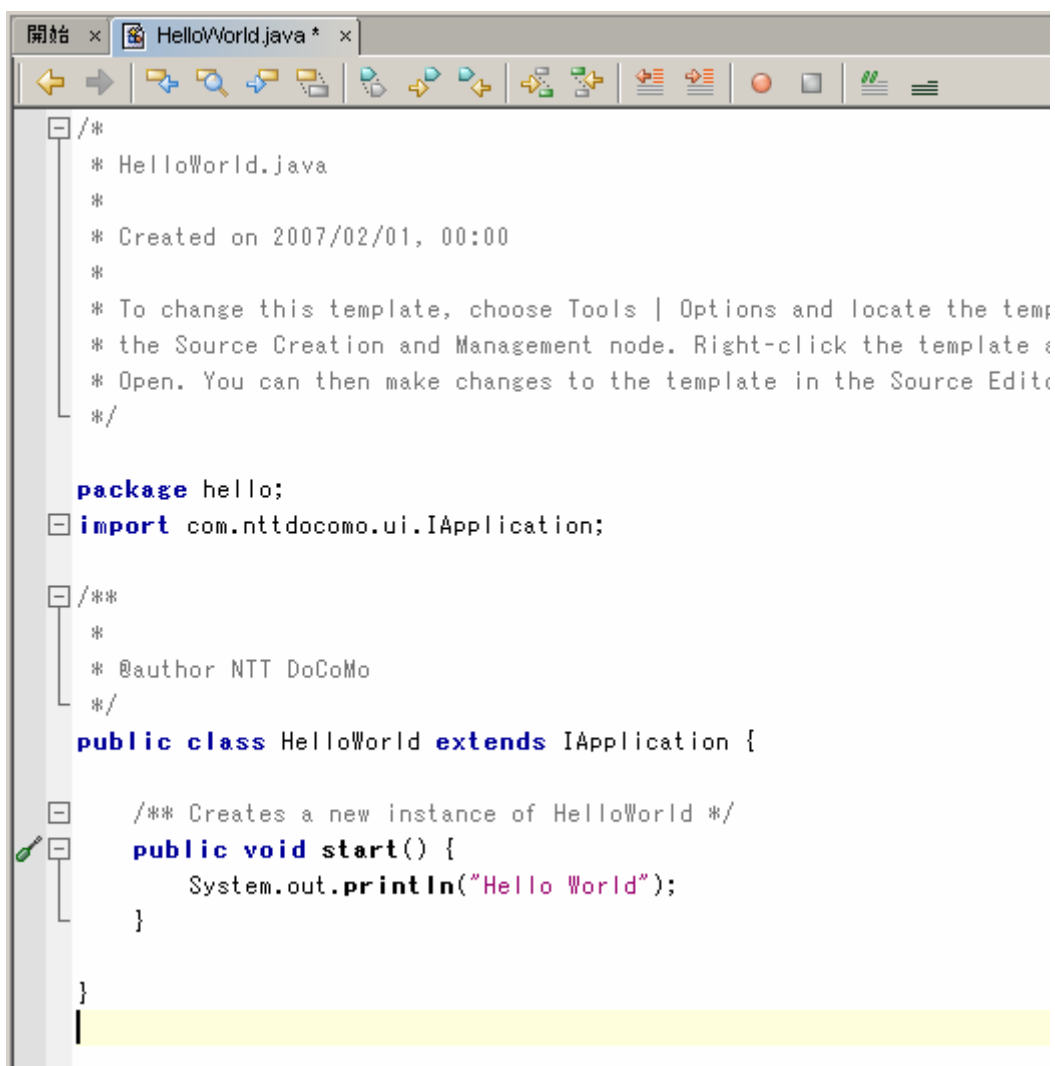


図 1 4-1 5 ソースファイルの編集

14.3.3. アプリケーションディスクリプタファイル(ADF)の設定

1. プロジェクトを選択して、「DoJa-5.0」→「ADF/トラステッド動作設定」を選択します（プロジェクトを選択していないと選べません）。
2. ADF/トラステッド動作設定画面が表示されるので、「ADF設定」タブを選択して、**AppName と AppClass を設定します**。（AppSize と LastModified は自動的に設定されますが、AppName と AppClass は手動で設定する必要があります。）
3. 必要に応じて各キーの属性値を変更してください（図 14-16）。
4. 「了解」を押すと保存されます。

ADFキー	値
AppName	HelloWorld
AppVer	
PackageURL	hello.jar
AppSize	
ConfigurationVer	
ProfileVer	
SPsize	
AppClass	hello.HelloWorld
AppParam	
LastModified	
UseNetwork	
TargetDevice	
LaunchAt	
MyConcierge	
UseTelephone	
UseBrowser	
UseDTV	
LaunchByMail	
LaunchByBrowser	
LaunchByToruCa	
LaunchByBML	
AllowPushBy	
AppTrace	
DrawArea	
GetSysInfo	

AppName と AppClass は、必ず手動で設定してください。

了解 取消し

図 14-16 ADF設定画面

各ADFの設定項目については、「4.7. アプリケーションディスクリプタファイル(ADF)の編集 表 4-3 アプリケーションディスクリプタファイル(ADF)の設定項目」を参照してください。

14.3.4. トラステッド動作の設定

トラステッド動作の設定をするには、次の手順に従います。

1. 「DoJa-5.0」→「ADF/トラステッド動作設定」を選択します。
2. ADF/トラステッド動作設定画面が表示されるので、「トラステッド動作設定」タブを選択して、必要に応じて各キーの属性値を変更してください（図 14-17）。
3. 「了解」を押すと保存されます。

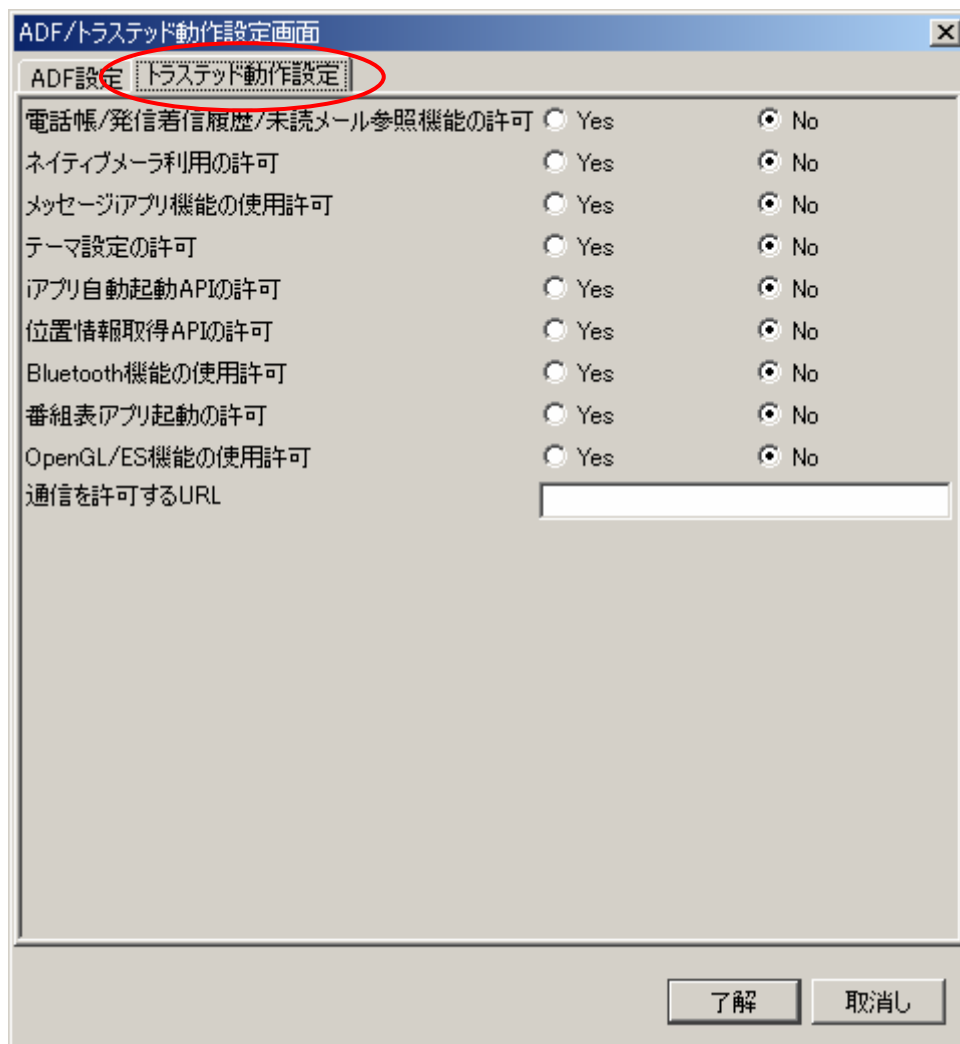


図 14-17 トラステッド動作設定画面

各トラステッド動作設定の設定項目については、「5.2 トラステッド動作の設定」を参照してください。

14.3.5. アプリケーションの構築と実行

アプリケーションを構築および実行するには、次の手順に従います。

1. プロジェクトウィンドウで、プロジェクトを選択します。
2. 右クリックメニューから「プロジェクトを構築」を選択します（図 14-18）。
3. プロジェクトがビルドされ、ビルドの経過が出力ウィンドウに表示されます。

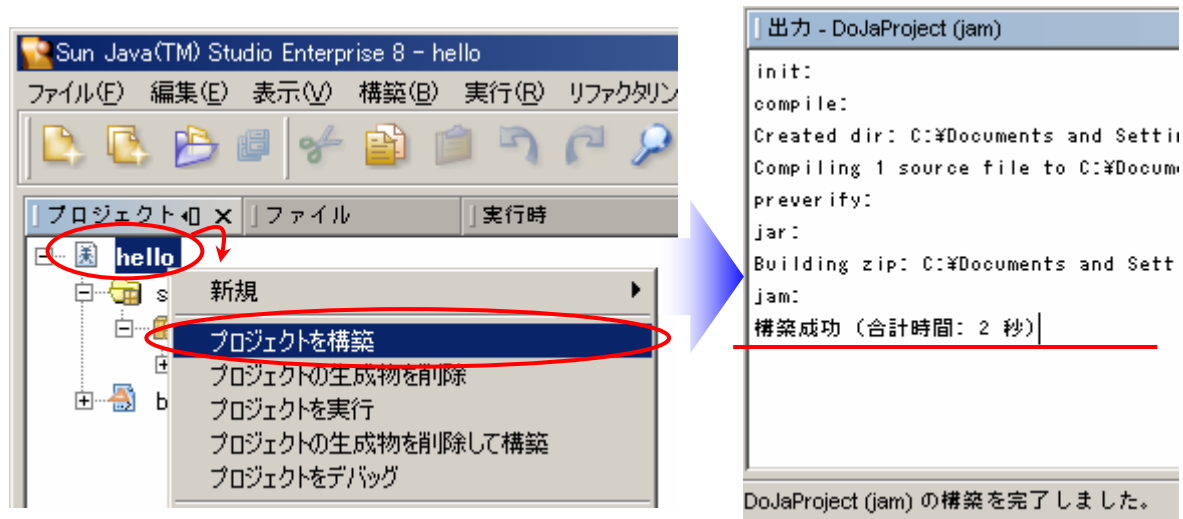


図 14-18 プロジェクト構築

4. プロジェクトのビルドに成功したら、右クリックメニューから「プロジェクトを実行」を選択します（図 14-19）。

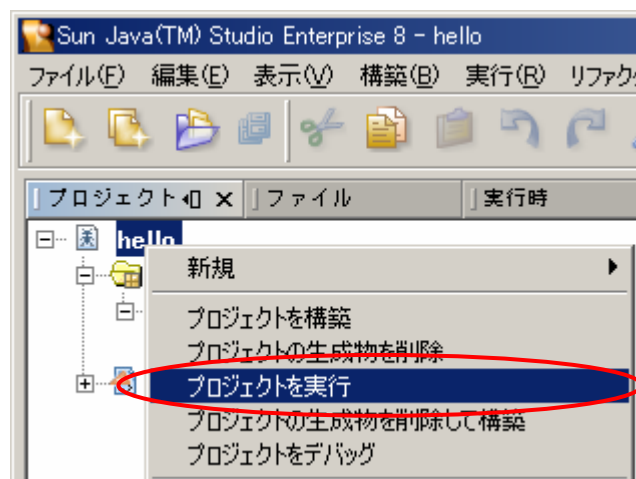


図 14-19 プロジェクト実行

5. 端末ウィンドウが表示されて、i アプリが実行されます（図 1 4-2 0）。



図 1 4-2 0 i アプリの実行

14.4. エミュレータの設定

14.4.1. デバイスの設定

エミュレータのデバイスを変更するには、次の手順に従ってください。

1. 「DoJa-5.0」→「起動設定」を選択すると、起動設定画面が表示されます（図 1 4-2 1）。
2. 「Device type」でエミュレーション端末の種類を選択します。
3. 「Launch type」で起動方法を選択します。
4. 「了解」を押します。



図 1 4-2 1 起動設定画面

14.4.2. ネットワークの設定

ネットワークを設定するには、次の手順に従ってください。

1. 「DoJa-5.0」→「ネットワーク設定」を選択すると、ネットワーク設定画面が表示されます（図 1 4-2 2）。
2. ネットワークの設定を変更して「了解」を押します。

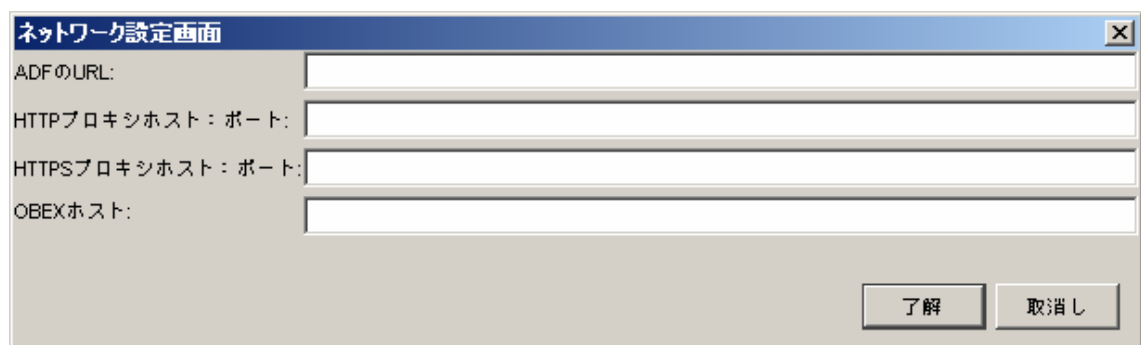


図 1 4-2 2 ネットワーク設定

ネットワークの各設定項目の詳細については、「5.1. ネットワークの設定」を参照してください。

14.4.3. IDE からモジュールを削除する

IDE から「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」を削除するには、次の手順に従います。

1. Sun Java Studio Enterprise もしくは、NetBeans IDE を起動します。
2. 「ツール」→「オプション」を選択します（図 14-23）。
3. 「IDE 構成」→「システム」→「モジュール」→「DoJa-5.0」を選択します。
4. 右クリックメニューから「削除」を選択します。

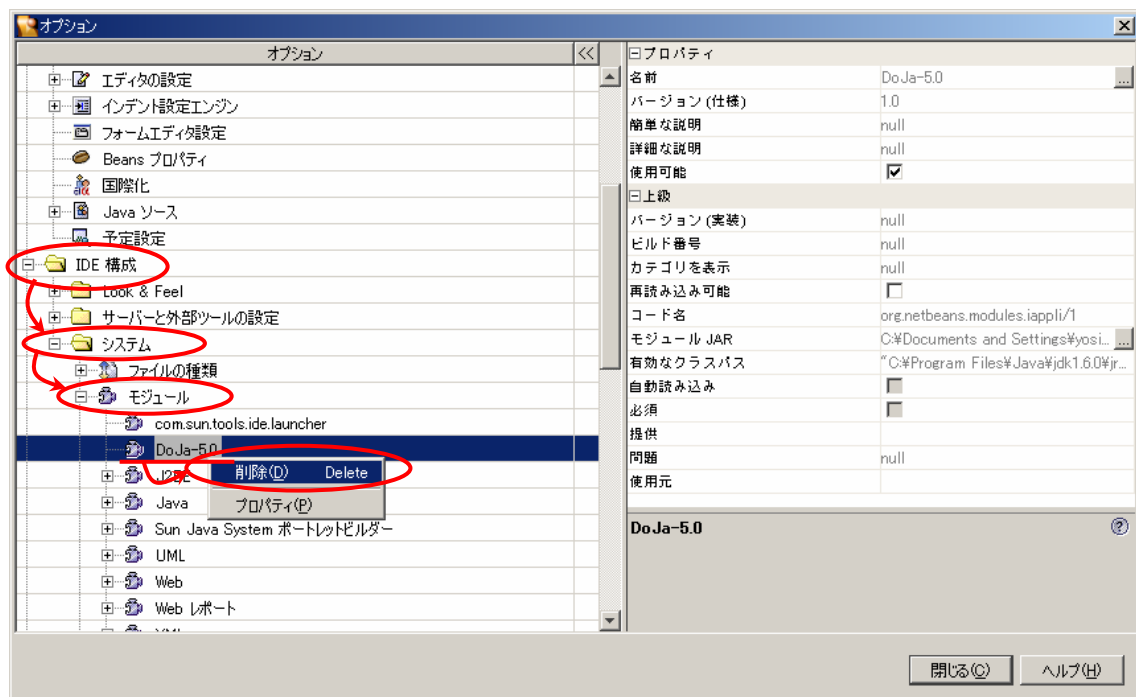


図 14-23 モジュールの削除

5. 削除の確認ダイアログが表示されるので、「はい」を押します（図 14-24）。
6. 「Sun Java Studio Enterprise 8 & NetBeans IDE 4.1 モジュール for iα ppli」が削除され、「DoJa-5.0」メニューが消えます。

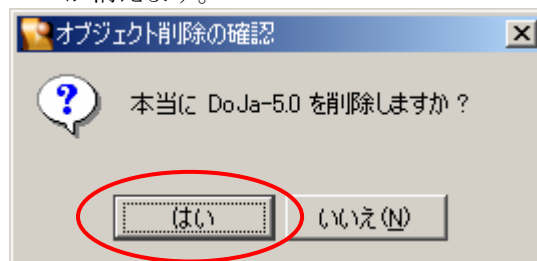


図 14-24 モジュールの削除確認

第15章. Eclipse の利用

この章では、Eclipse と「Eclipse プラグイン for iα ppli」の拡張モジュールについて説明します。

15.1. Eclipse について

Eclipse に「Eclipse プラグイン for iα ppli」をインストールする事で、Eclipse からエミュレータを直接起動することができます。「Eclipse プラグイン for iα ppli」は、Eclipse で i アプリを作成するためのプラグインです。

「Eclipse プラグイン for iα ppli」は、Eclipse3.0 および Eclipse3.1 で開発しています。そのため「Eclipse プラグイン for iα ppli」は、Eclipse3.0 か Eclipse3.1 で使われる事を推奨します。

Eclipse は「Language Pack」をダウンロードして、Eclipse のインストール（解凍）先に上書きする事で日本語に対応する事ができます。

Eclipse のダウンロードや、Eclipse の使用とインストールに関する詳細なドキュメントは、Eclipse の Web サイト <http://www.eclipse.org/> を参照してください。

注意：

Eclipse のインストール先には、「スペース（空白）」及び「2 バイト文字（全角文字）」は含めないようにしてください。

また、古いプラグインや他のプロファイルのプラグインが入っている場合は上書きされます。もし、DoJa-3.5 や DoJa-4.0/4.1 と共存させたい場合は、別の Eclipse 環境 を用意してください。

Eclipse で作成された i アプリ (Jar ファイル) にはデバッガ情報が含まれています。そのため、i モード携帯電話にダウンロードする i アプリは、なるべくエミュレータの「iα ppliTool」を使ってビルドしなおしてください。

Eclipse で i アプリにデバッグ情報を含めないようにするには、「プロジェクト」→「プロパティ」→「Java コンパイラ」のクラス・ファイル生成で「デバッガーによって使用される」チェックを全て外すことで可能です。

ただし、デバッグができなくなりますので、あまりお勧めできません。

15.2. 「Eclipse プラグイン for iα ppli」のインストール

本節では、Eclipse 用の拡張モジュールをインストールする手順を説明します。

15.2.1. 始める前に

Eclipse で「Eclipse プラグイン for iα ppli」を使用するには、Java 2 Platform, Standard Edition SDK (J2SE SDK) と Eclipse をインストールしてから、「Eclipse プラグイン for iα ppli」をインストールする必要があります。

- J2SE SDK は、次の Web サイトからダウンロードできます。
<http://java.sun.com/j2se/>
 - J2SE SDK のインストール後は、環境変数の PATH に J2SE SDK の bin を追加してください (コントロールパネル → システム → 詳細 → 「環境変数」)。
これは、iα ppliTool では自動的に検出しています。
- Eclipse と、その Language Pack は、次の Web サイトからダウンロードできます。
<http://www.eclipse.org/>

15.2.2. インストール

「Eclipse プラグイン for iα ppli」をインストールするには、2.2. インストール手順の セットアップタイプの選択画面 (図 2-6) で、「カスタム」を選択し、機能の選択画面で、「Eclipse3.0/3.1 プラグイン」をチェックします (図 15-1)。

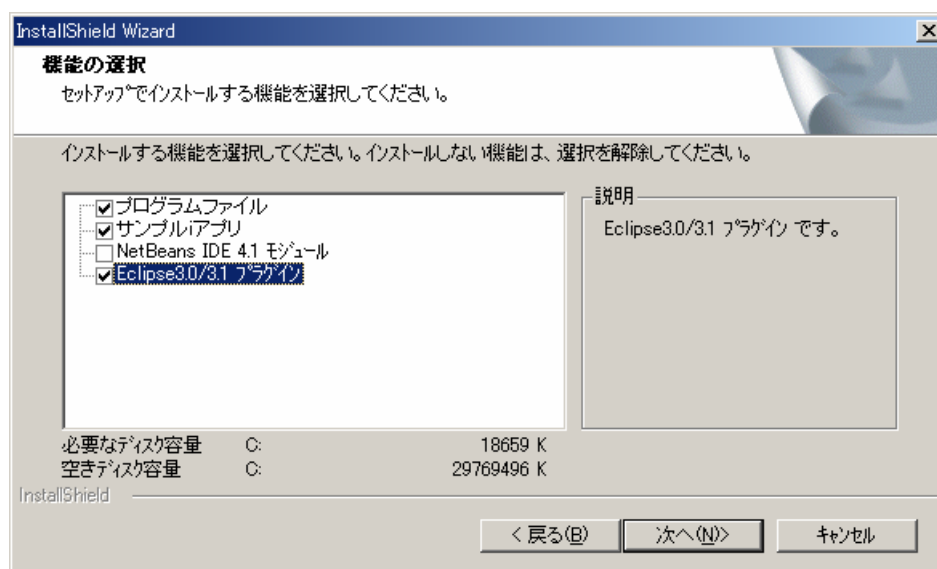


図 15-1 Eclipse プラグインのインストール

Eclipse は、インストーラが無いので、どこにあるかを指定する必要があります。

「Eclipse3.0/3.1 プラグイン」をチェックして、「次へ」を押すと、Eclipse のインストール先ディレクトリを指定する画面が表示されます。

15.3. 新規アプリケーションの作成

この節では、新しいアプリケーションの作成方法について説明します。

15.3.1. DoJa を利用する前に

Eclipse で DoJa を利用できるようにするためには、Eclipse に「iα ppli Development Kit for DoJa-5.0」のインストール先を設定しておく必要があります。

「Eclipse プラグイン for iα ppli」を組み込んだ状態で、最初に設定されている「iα ppli Development Kit for DoJa-5.0」のインストール先は、「C:\iDKDoJa5.0」になっています。Eclipse に「iα ppli Development Kit for DoJa-5.0」のインストール先を設定するには、次の手順に従います。

1. 「ウィンドウ」→「設定」を選択します。
2. 設定ウィンドウが開くので、設定ウィンドウから「ワークベンチ」→「DoJa-5.0 Environment」を選択すると、「iα ppli Development Kit for DoJa-5.0」のインストール先を設定する画面になります。

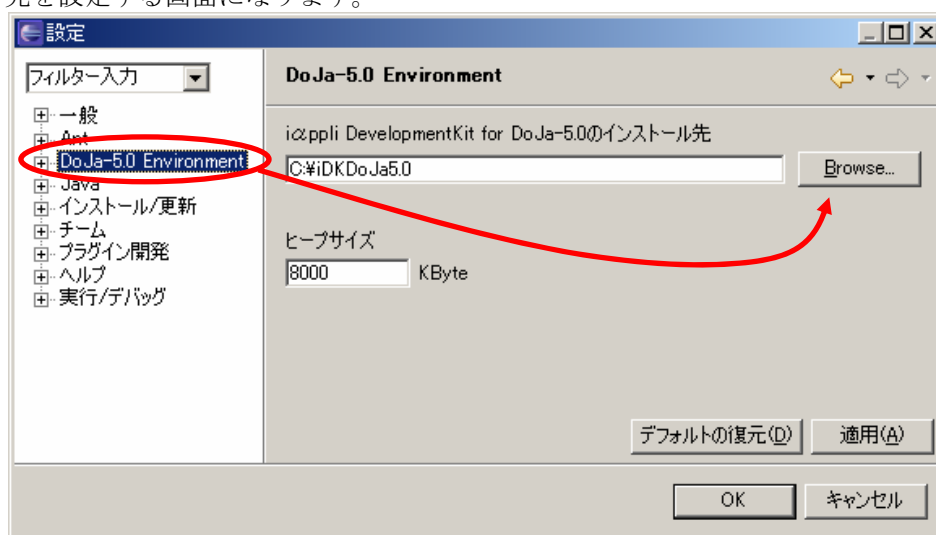


図 1 5 - 2 Eclipse 設定ウィンドウ

3. 「Browse(B)...」を押して「iα ppli Development Kit for DoJa-5.0」のインストール先を設定し、「OK」を押します。

15.3.2. プロジェクトメニューのカスタマイズ

Eclipse で DoJa の設定を行えるようにするには、次の手順が必要になります。

新たにプラグインをインストールしなおした場合にも必要です。

1. 「ウィンドウ」→「パースペクティブのカスタマイズ」を選択します。
2. 「コマンド」タブ→「DoJa-5.0 設定」にチェックを付けます。
一度チェックをすると、次回の Eclipse 起動時にはチェックされた状態になります。

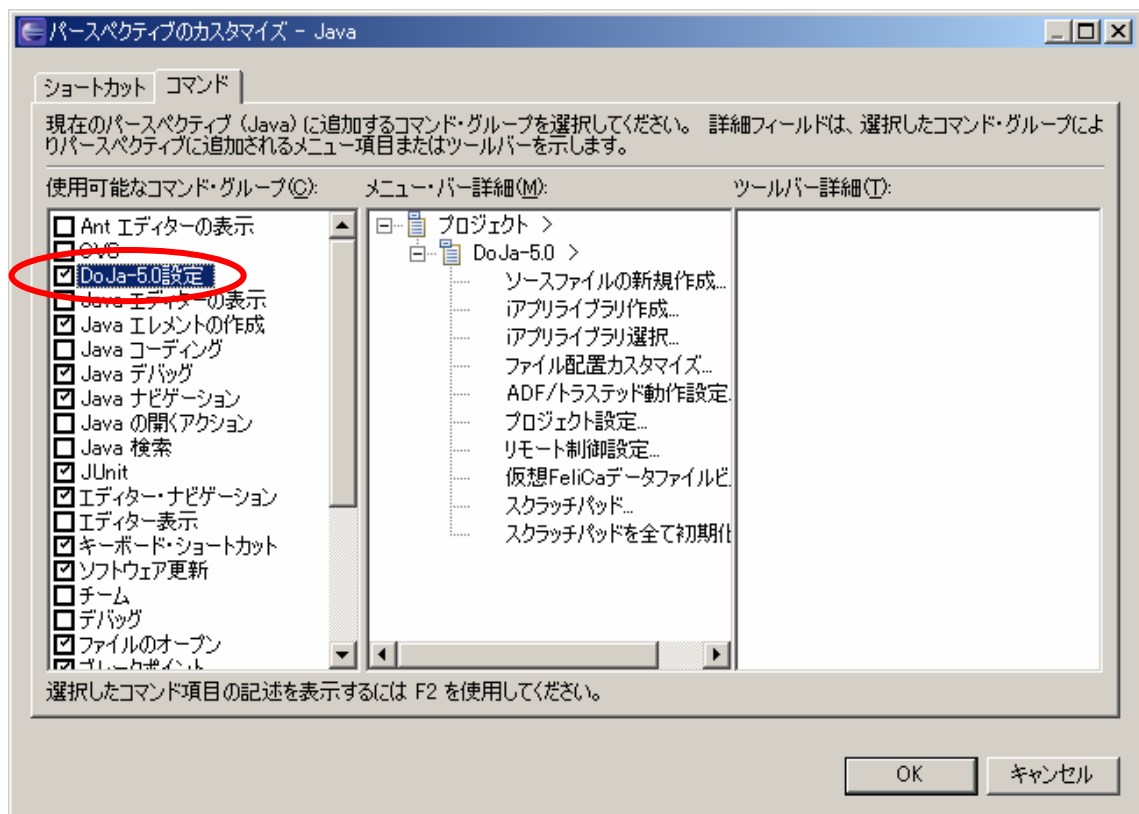


図 15-3 プロジェクトメニューのカスタマイズ

15.3.3. プロジェクトの作成

アプリケーションのソースファイルと属性ファイルを収容するプロジェクトを作成します。プロジェクトを作成するには、次の手順に従います。

1. 「ファイル」→「新規」→「プロジェクト」を選択します。
2. 新規プロジェクトウィンドウが表示されるので、「Java」の「DoJa-5.0 プロジェクト」を選択し、「次へ」を押します。

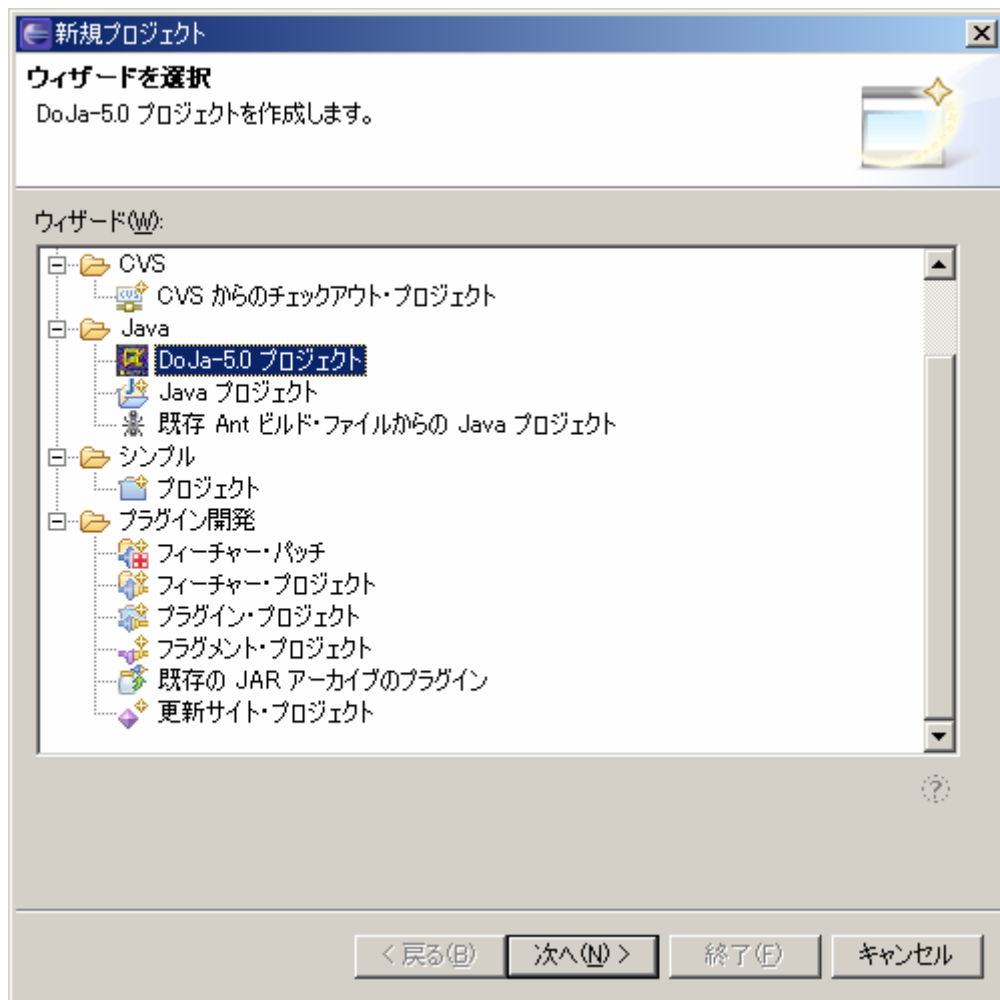


図 1 5-4 新規プロジェクトウィンドウ

3. プロジェクト名設定ウィンドウが表示されるので、プロジェクト名を入力して「次へ」を押します。
たとえば、hello と入力します。

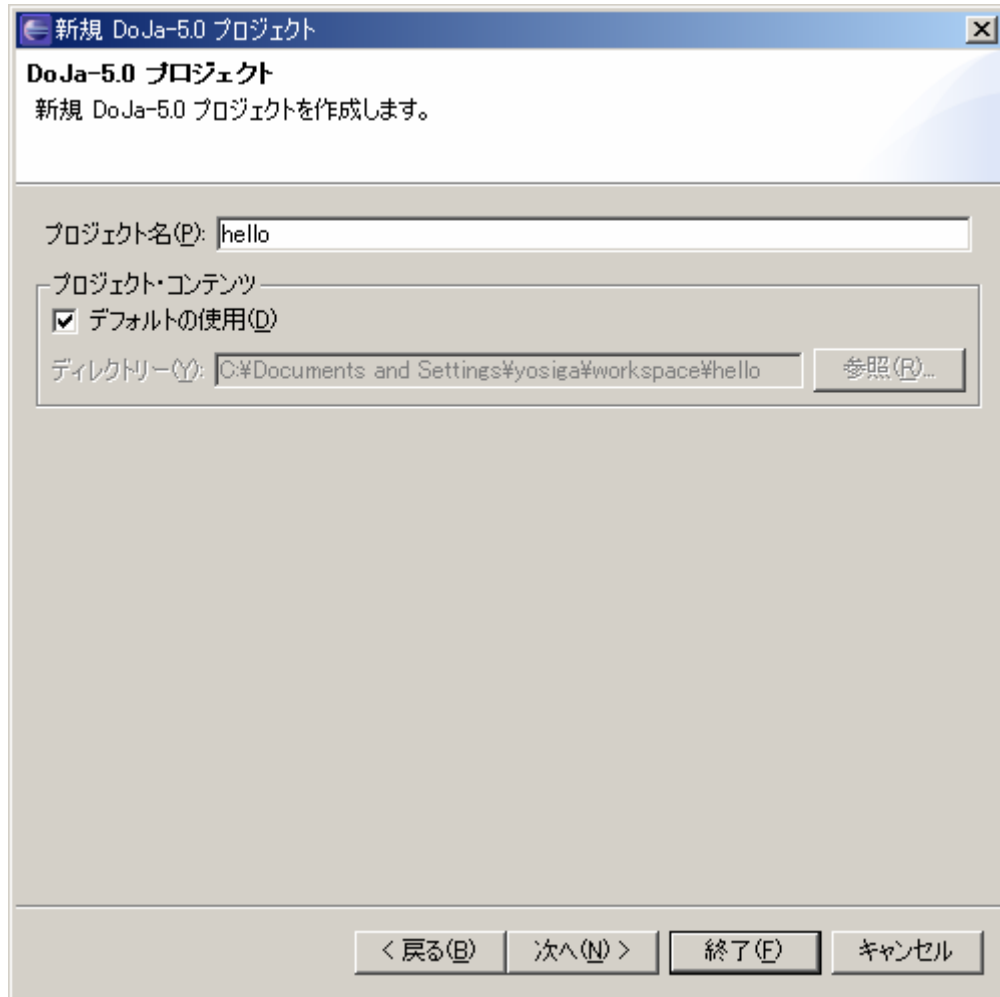


図 1 5-5 プロジェクト名設定ウィンドウ

- 「プロジェクトの内容」の「デフォルトの使用」のチェックを外す事で、既に「i alpha Development Kit for DoJa-5.0」で作成されたプロジェクトを「ディレクトリ」で指定する事もできます。

4. Java 設定ウィンドウが表示されるので、もしビルド環境を変更したい場合は、設定内容を変更して「終了」を押します。
(ここでは、何も変更せずに「終了」を押します。)

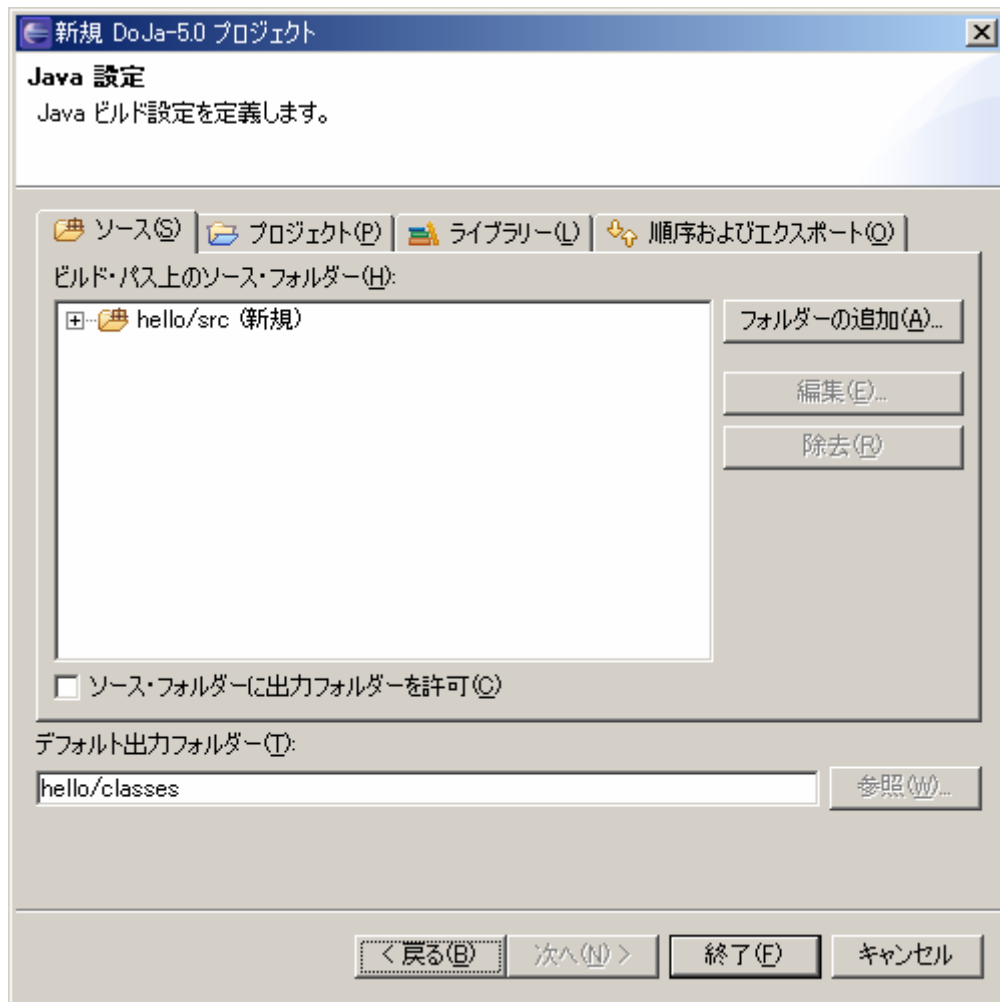


図 1 5-6 Java 設定ウィンドウ

5. ナビゲーターウィンドウにプロジェクト名 (hello) が追加されます。

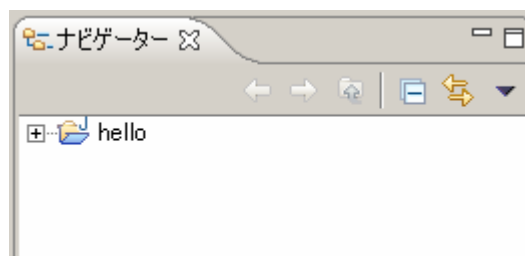


図 1 5-7 ナビゲーターウィンドウ

15.3.4. ソースファイルの作成

アプリケーションのソースファイルを作成するには、次の手順に従います。

1. パースペクティブの選択から「Java」パースペクティブを選択します。

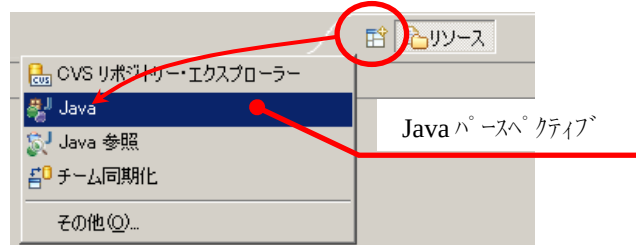


図 1 5-8 Java パースペクティブの選択

2. パッケージ・エクスプローラーウィンドウに切り替わるので、プロジェクト(hello)→「src」をクリックし、マウスの右ボタンから「新規」→「クラス」を選択します(図 1 5-9)。

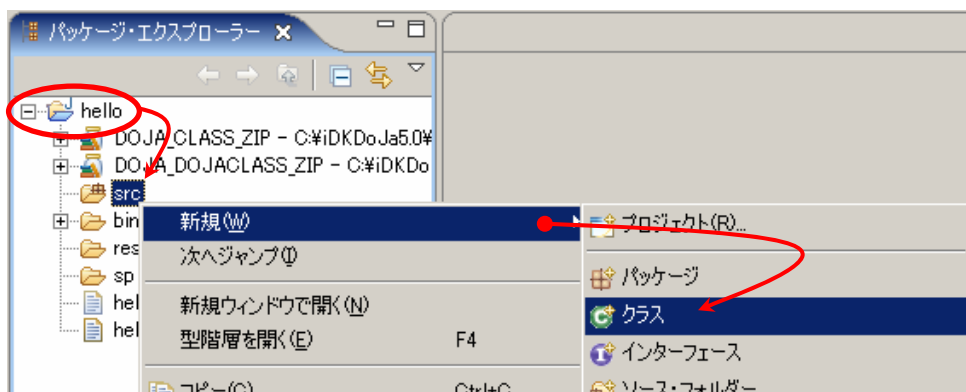


図 1 5-9 新規作成クラスの選択

3. 新規クラス設定ウィンドウが表示されるので、パッケージ名とクラス名を入力し、「終了」を押します。
たとえばパッケージに `hello`、名前に `HelloWorld` と入力します。

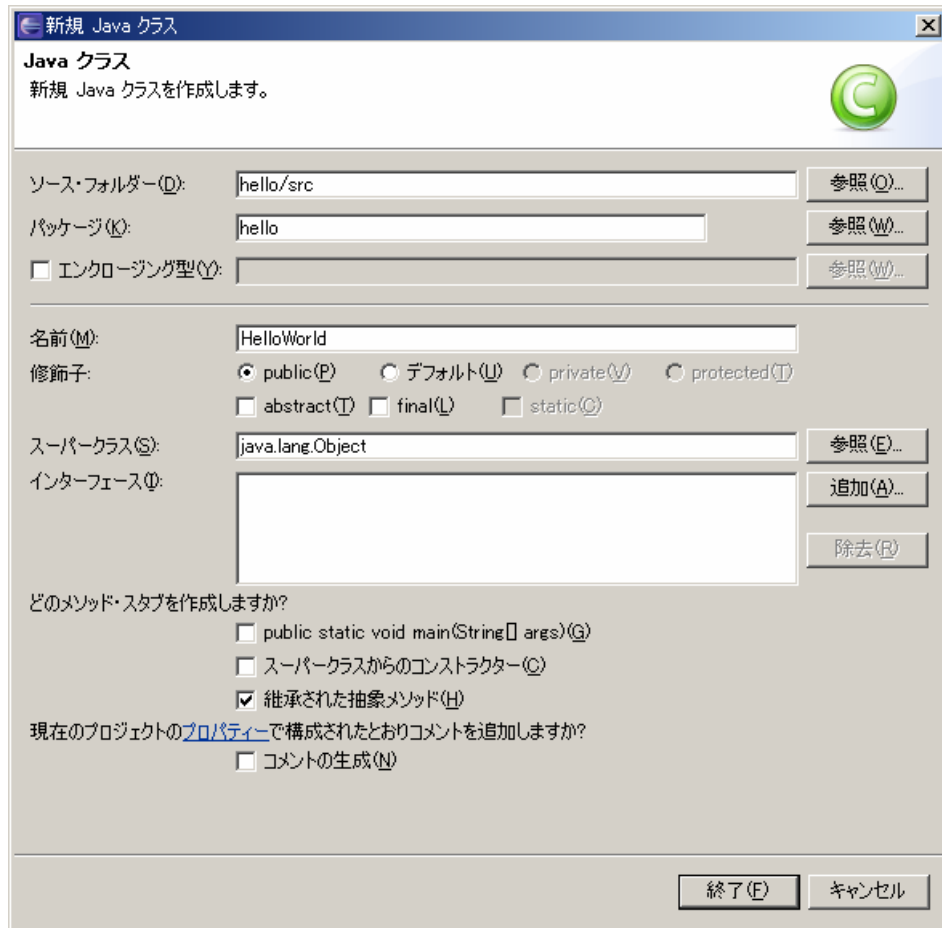


図 1 5 - 1 0 新規クラス設定ウィンドウ

4. ソースファイルを編集します。
ソースファイルは、1～3の手順以外に、マウスのドラッグ&ドロップで既存のソースを挿入する事もできます。
たとえば、ソースファイル中で
package hello; の後に
import com.nttdocomo.ui.IApplication;
public class HelloWorld に続けて、
extends IApplication を追加し、
public class HelloWorld extends IApplication { の後に
public void start() {
 System.out.println("Hello World");
}
を追加します（図 1 5-1 1）。

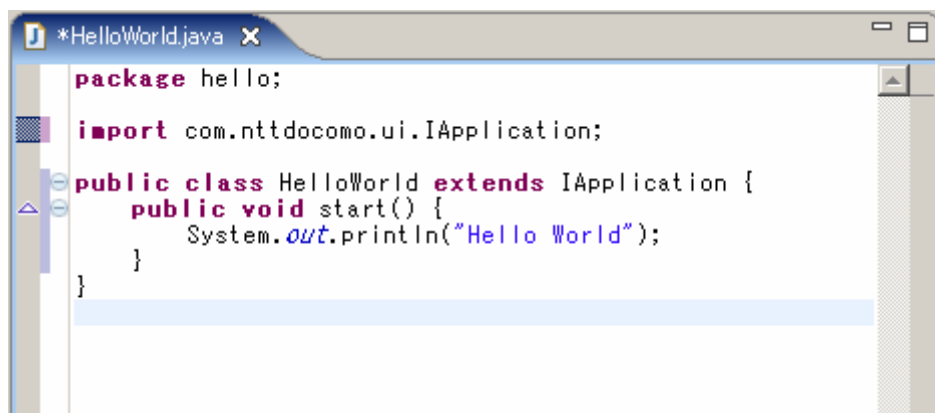


図 1 5-1 1 ソースファイルの編集

5. ソースファイルを保存します。
「ファイル」→「すべて保管」の順でメニュー項目を選択するか、**Ctrl-S** を押して、ソースファイルを保存します。
この時点で、ソースファイルは自動的にコンパイルされます。
もし、ソースファイルの横に  や  が表示されたら、その行かその行に関連する処理に誤りがあります。マークの上にカーソルを置くとその理由が表示されますのでソースを修正してください。

15.3.5. プロジェクト設定からのソースファイルの作成

アプリケーションのソースファイル作成は、次の手順から行えます。

ここでは、テンプレートから必要なリスナを自動で組み込みます。

1. パッケージ・エクスプローラーウィンドウから**プロジェクト (hello)** をクリックして、「プロジェクト」→「DoJa-5.0」→「ソースファイルの新規作成」をクリックします。
(※ **プロジェクトを選んでいないと選択できません。**)
2. プロジェクト設定ウィンドウが表示されるので、「ソースファイルの新規作成」タブから、必要に応じて各キーの属性値を変更してください。

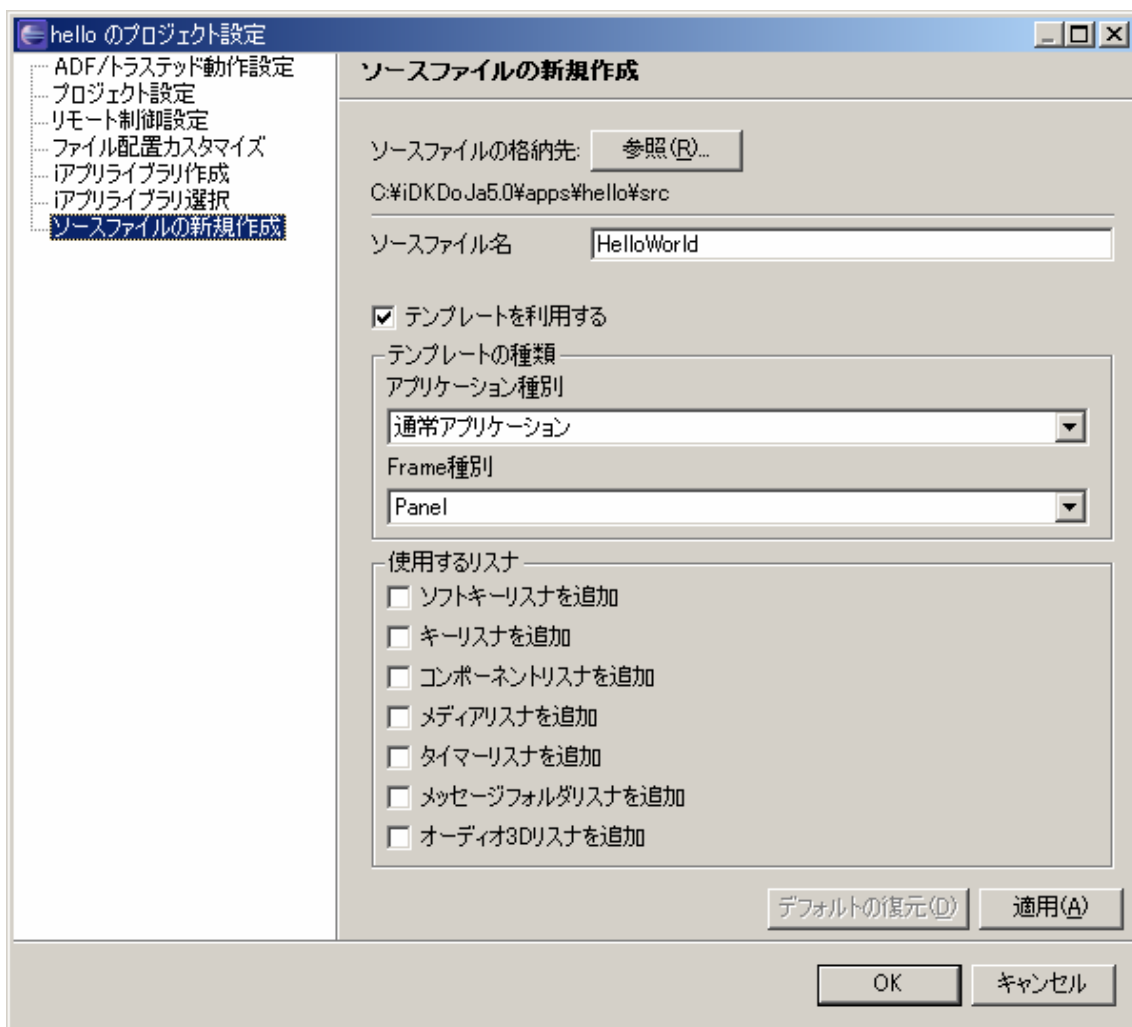


図 1 5 - 1 2 ソースファイルの新規作成画面

各ソースファイルの新規作成の設定項目については、「4.2.1 ソースファイルの作成」を参照してください。

15.3.6. アプリケーションディスクリプタファイル(ADF)設定

ADFの設定をするには、次の手順に従います。

1. パッケージ・エクスプローラーウィンドウから**プロジェクト (hello)** をクリックして、「プロジェクト」→「DoJa-5.0」→「ADF/トラステッド動作設定」をクリックします。
(※プロジェクトを選んでいないと選択できません。)
2. プロジェクト設定ウィンドウが表示されるので、「ADF 設定」タブから、必要に応じて各キーの属性値を変更してください。

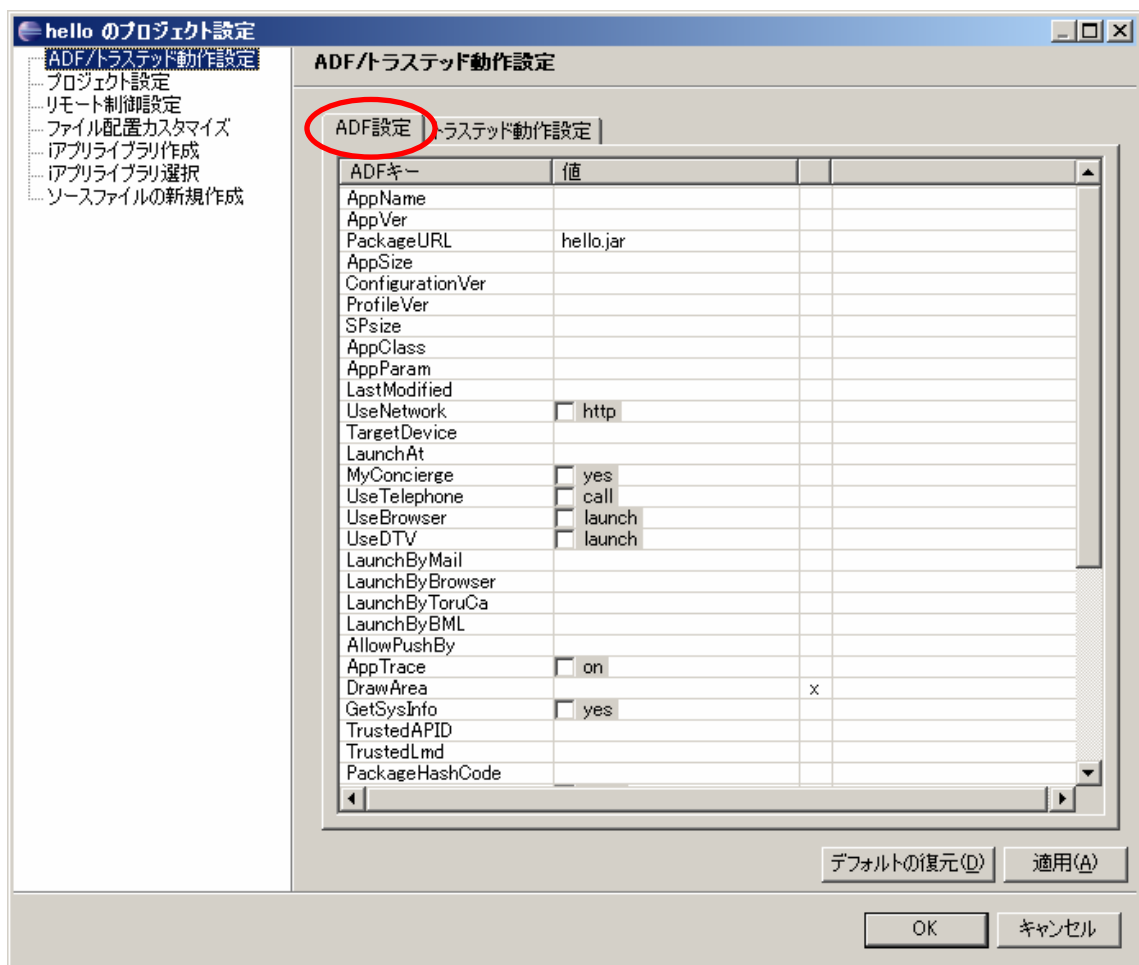


図 15-13 ADF 設定画面

各ADFの設定項目については、「4.7. アプリケーションディスクリプタファイル (ADF) の編集 表 4-3 アプリケーションディスクリプタファイル(ADF)の設定項目」を参照してください。

15.3.6.1. FeliCaCPID の設定

ADF の FeliCaCPID は ADF の URL からハッシュ値を作成します。
ADF の FeliCaCPID 欄をクリックすると ADF の URL 画面が表示されるので、実際にダウンロードする際の ADF の URL をスキーム (http://) から入力して「設定」を押します (図 15-14)。



図 15-14 FeliCaCPID の設定

FeliCaCPID を削除したい場合は、ADF の URL を空にしてください。

15.3.6.2. FeliCa フリー領域の関連付け

ADF に FeliCaCPID と FeliCaPIN が設定されている場合、自動的に仮想 FeliCa のフリー領域に FeliCaCPID と FeliCaPIN を関連付ける事ができます。

ADF に FeliCaCPID と FeliCaPIN が設定されていると ADF 設定画面の「OK」を押したタイミングでどのフリー領域に関連付けるかを問い合わせます (図 15-15)。

一度、問い合わせると変更しない限り 2 回目以降は問い合わせません。



図 15-15 フリーエリアの選択

対象となるフリー領域を選択して「設定」を押すと、そのフリー領域に関連付けられます。

15.3.8. アプリケーションの構築と実行

アプリケーションの構築および実行をするには、次の手順に従います。

1. 「実行」→「構成および実行」(もしくは「実行」)を押します。
2. 起動構成ウィンドウが表示されるので、起動構成の「DoJa-5.0 アプリケーション」をマウスでダブルクリック、もしくは、「新規」ボタンを押します。
3. 「新規構成」が追加されるので、これを選択します。
「新規構成」の名前は、必要に応じてアプリケーション名などに変更してください。
4. 「Project」で実行するアプリケーションのプロジェクトを選択します。
5. 「Device type」でエミュレーション端末の種類を選択します。
6. 「Launch type」で起動方法を選択します。
7. 「エントリー ID」で、位置情報起動、位置情報画像起動、位置情報電話帳時のエントリー IDを指定します。
8. 「アプリケーション起動パラメータ」で、外部機器起動のパラメータを指定します。
また、BML 起動・DTV 起動時のパラメータも指定可能です。
BML 起動の場合、ここで指定した値が `IApplication#getParameter("bml_param")` で取得され、DTV 起動の場合、ここで指定した値が `IApplication#getParameter("service_id")` で取得されます。
9. 「Trace Setting」で、出力したいトレースログを選択します。(通常は選択不要)
10. 「実行」を押します。

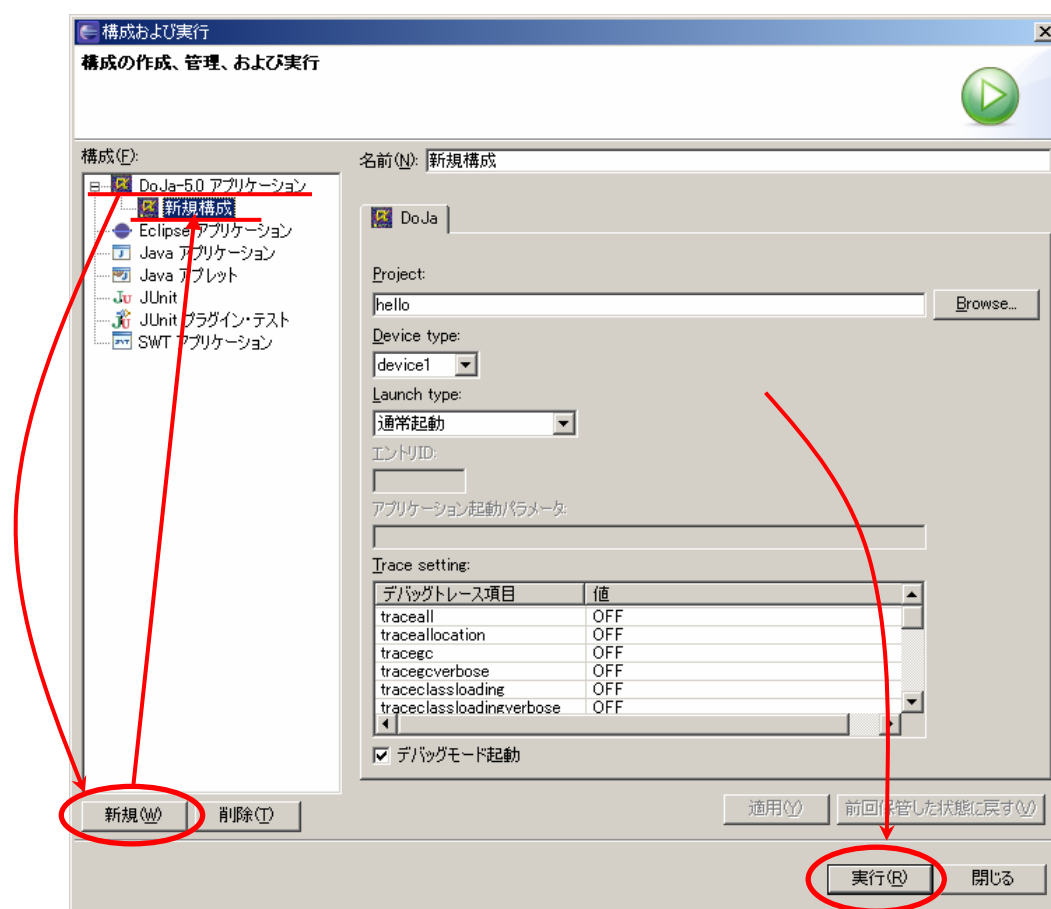


図 15-17 起動構成ウィンドウ

11. preverify（事前検証）、Jar ファイルの作成後に端末ウィンドウが起動し、アプリケーションが実行されます。
12. コンソールに出力された内容は、Eclipseのコンソールに出力されます（図 15-18）。

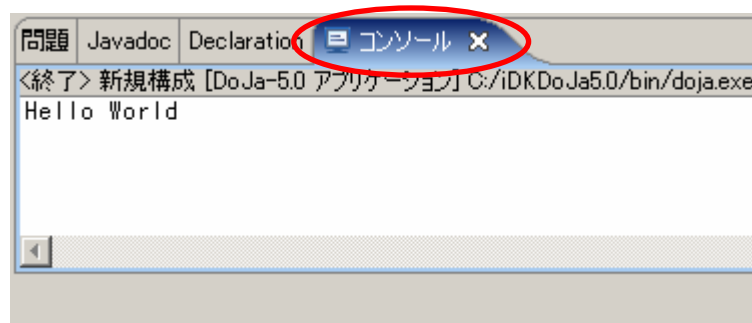


図 15-18 コンソールに出力された内容の表示

※ Eclipse プラグインにつきましては、ネイティブアプリケーション表示・連携起動機能（4.1 6 メーラ起動、4.1 7 ブラウザ起動、4.1 8 コードリーダー起動、4.2 3 トルカ起動、4.2 5 BML 起動、4.2 6 D T V 起動）はありません。そのため、メーラ起動・ブラウザ起動・トルカ起動・BML 起動・D T V 起動につきましては、通常起動と同様に「実行」ボタン押下後即エミュレータが起動します。（起動タイプは「Launch type」で選択した起動方法が反映されます。）

15.3.9. プロジェクトの設定

プロジェクトの設定をするには、次の手順に従います。

1. 「プロジェクト」→「DoJa-5.0」→「プロジェクト設定」を選択します。
2. プロジェクト設定ウィンドウが開きます（図 1 5 - 1 9）。

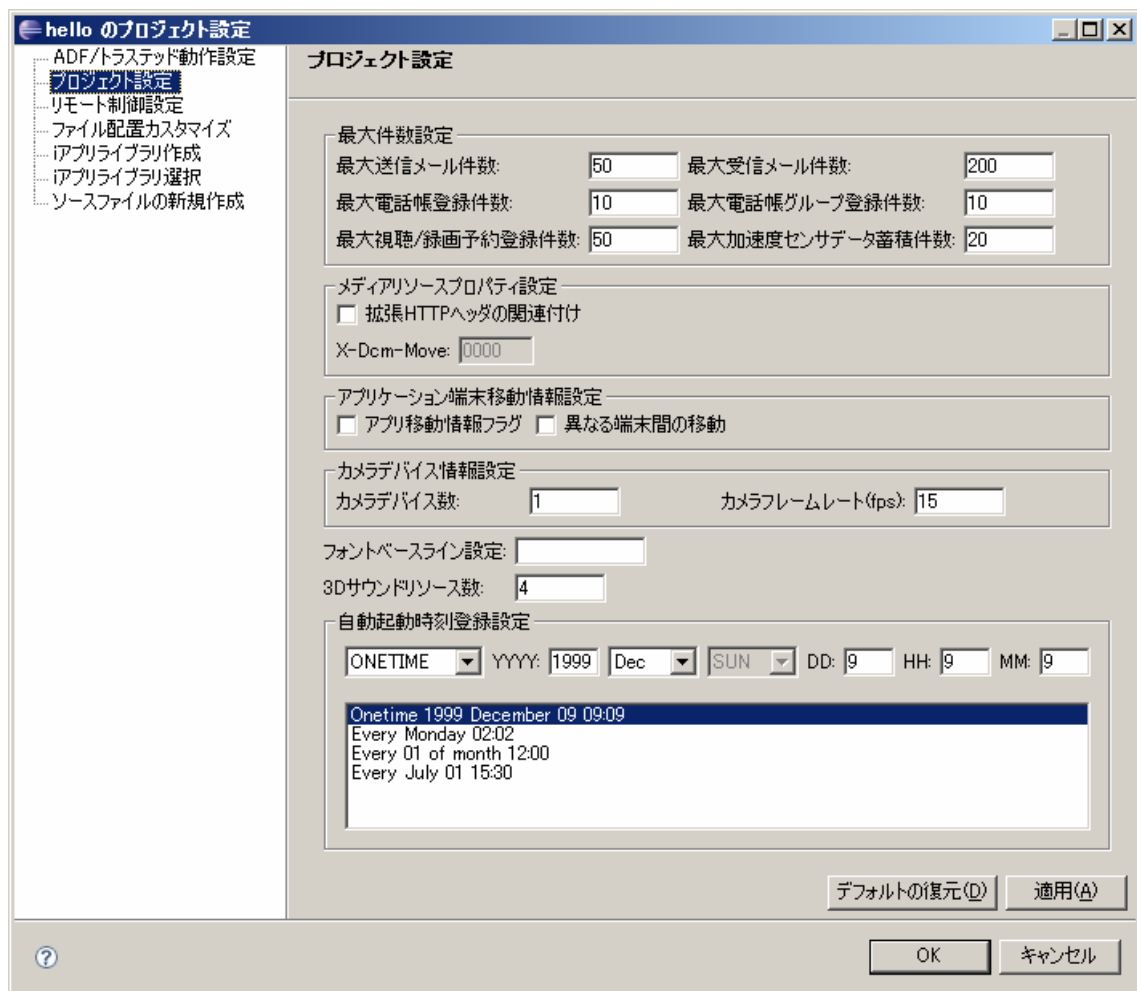


図 1 5 - 1 9 プロジェクト設定画面

各プロジェクト設定の設定項目については、「4.5. プロジェクトの設定」を参照してください。

15.3.10. リモート制御の設定

リモート制御の設定をするには、次の手順に従います。

1. 「プロジェクト」→「DoJa-5.0」→「リモート制御設定」を選択します。
2. プロジェクト設定ウィンドウが開きます。（図 15-20）。

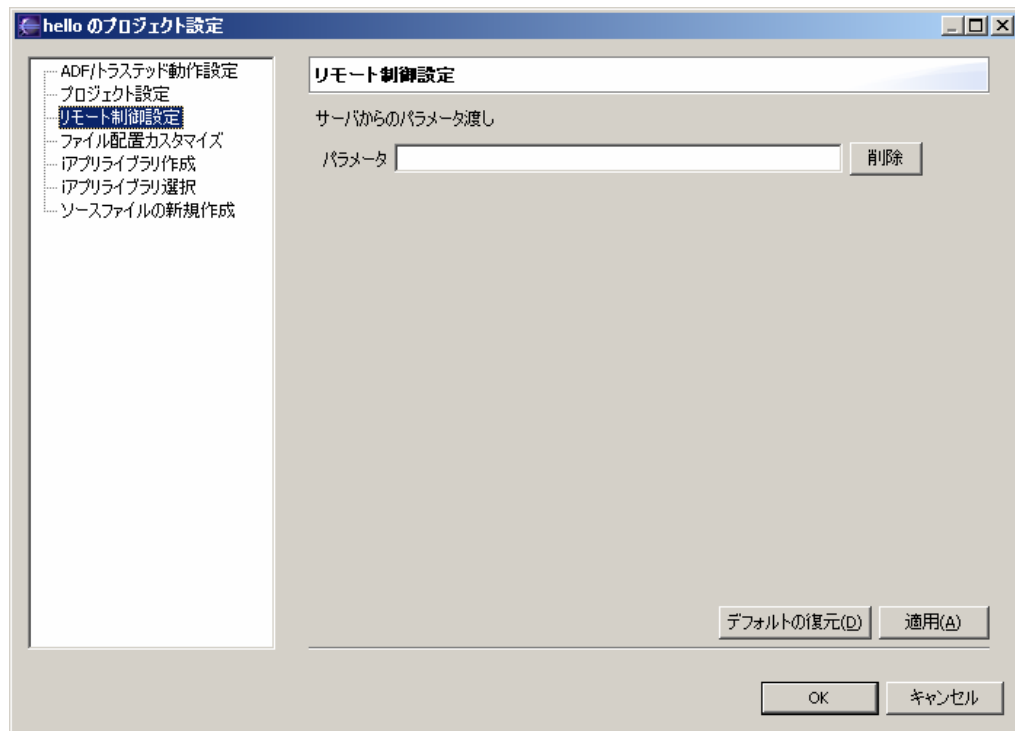


図 15-20 リモート制御設定画面

3. テキストフィールドにパラメータを入力し、「適用」、または「OK」を押します。
「アプリケーション名.properties」ファイルにパラメータが追加されます。
4. 「削除」ボタンを押した直後に「適用」ボタン、または「OK」ボタンを押すと、「アプリケーション名.properties」ファイルからパラメータが削除されます。

設定したパラメータをアプリケーションから取得する方法については、「5.4 リモート制御の設定」を参照してください。

15.3.11. ファイル配置カスタマイズ

ファイル配置カスタマイズの設定をするには、次の手順に従います。

1. 「プロジェクト」→「DoJa-5.0」→「ファイル配置カスタマイズ」を選択します。
2. プロジェクト設定ウィンドウが開きます（図 15-21）。

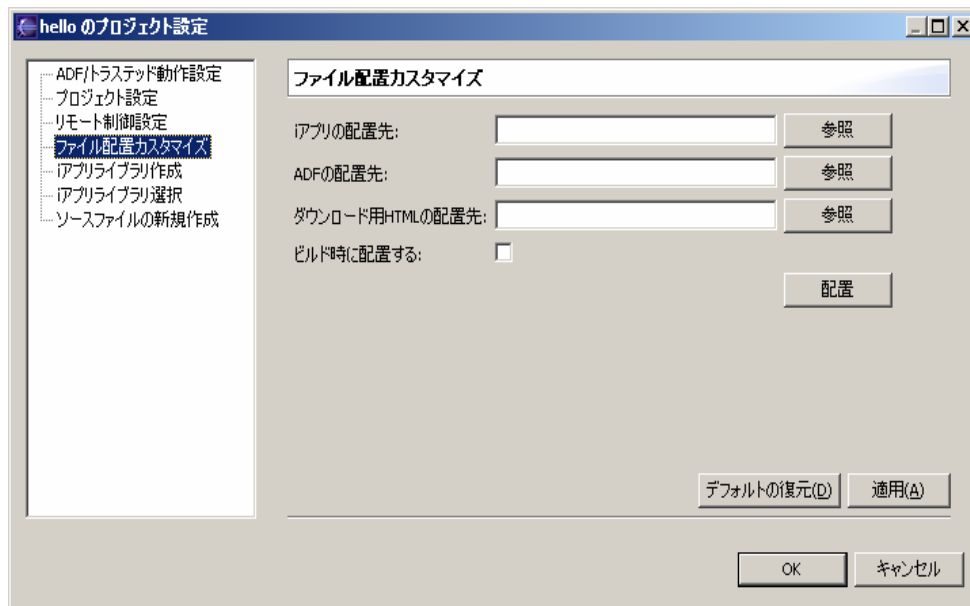


図 15-21 ファイル配置カスタマイズ

3. それぞれの配置先を入力し、「適用」または「OK」を押します。
このとき、ファイル名まで指定するとそのファイル名に変更されます。
4. 「配置」を押すと、配置先にファイルがコピーされます。
5. 「ビルド時に配置する」にチェックをつけた場合、i アプリのビルド時に自動的に配置処理を実行します。

詳細については、「4.6. ファイル配置カスタマイズ」を参照してください。

15.3.12. iアプリライブラリの作成

i アプリライブラリの作成を設定するには、次の手順に従います。

1. 「プロジェクト」→「DoJa-5.0」→「i アプリライブラリ作成」を選択します。
2. プロジェクト設定ウィンドウが開きます（図 15-22）。

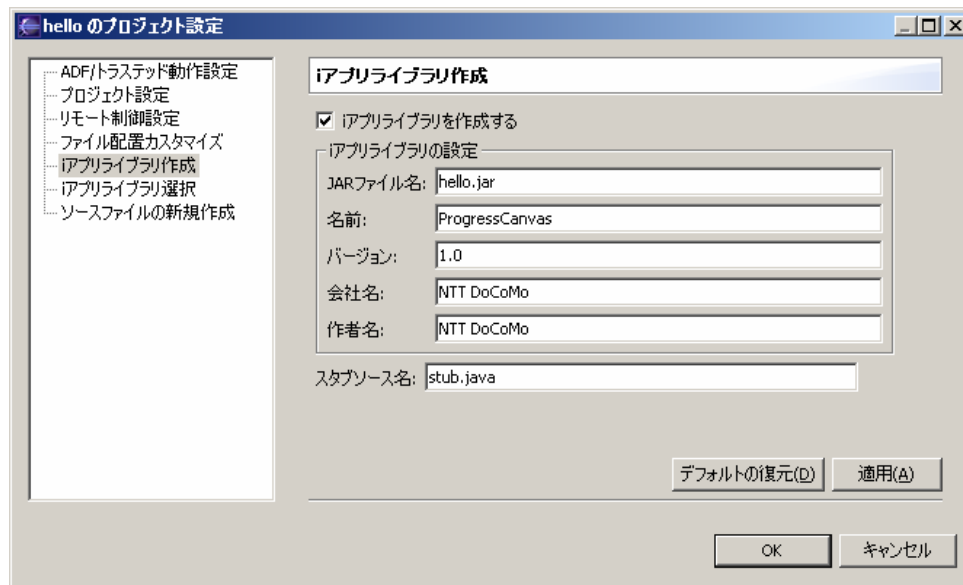


図 15-22 i アプリライブラリ作成

3. 「i アプリライブラリを作成する」にチェックをします。
4. このライブラリの属性（「Jar ファイル名」、「名前」、「バージョン」、「会社名」、「作者名」と「スタブソース名」（.java）を設定します。
スタブソース名は、ライブラリをテストするためのプログラムで、あらかじめ作成しておかなくてもテンプレートから自動的に作成されます。
5. 「OK」を押します。
これで i アプリライブラリが作成できるようになります。

詳細については、「6.7.1. i アプリライブラリの作成」を参照してください。

15.3.13. iアプリライブラリの利用

i アプリライブラリを利用するには、次の手順に従います。

1. 「プロジェクト」→「DoJa-5.0」→「i アプリライブラリ選択」を選択します。
2. プロジェクト設定ウィンドウが開きます（図 15-23）。

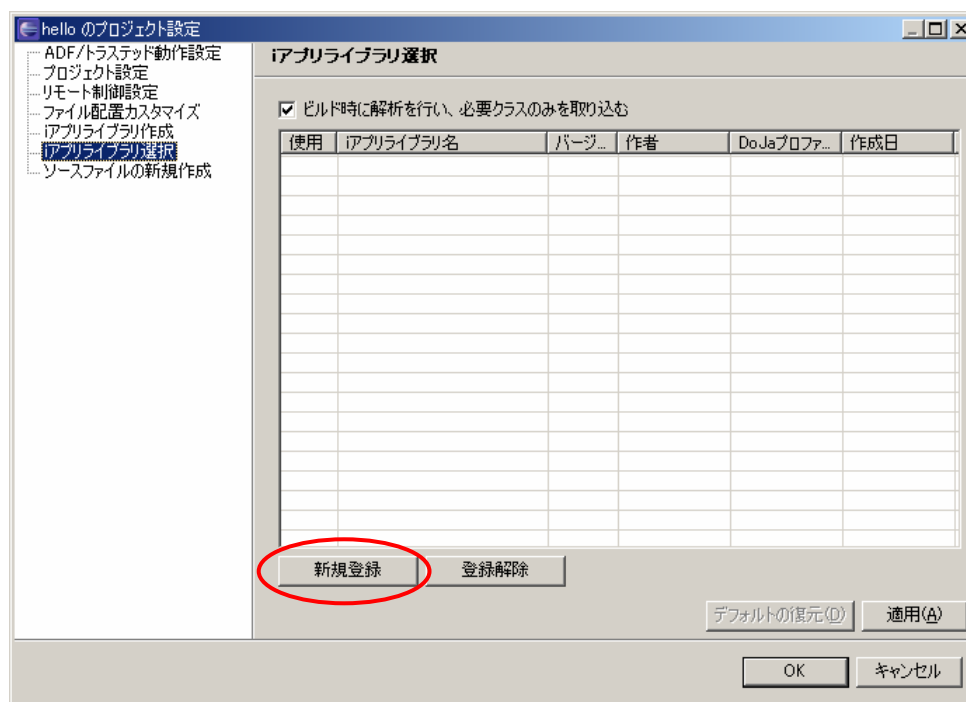


図 15-23 i アプリライブラリ選択

3. i アプリライブラリ選択画面が表示されるので「新規登録」を押します。
4. ファイル選択ダイアログで登録したい i アプリライブラリ(.jar)を選択すると i アプリライブラリの選択画面に追加されます(図 15-25)。
ここで、登録されたライブラリは、他のプロジェクトでも利用できます。
追加した i アプリライブラリは、右下の「キャンセル」を押しても消えませんが、削除したい場合は、「登録解除」で消してください。

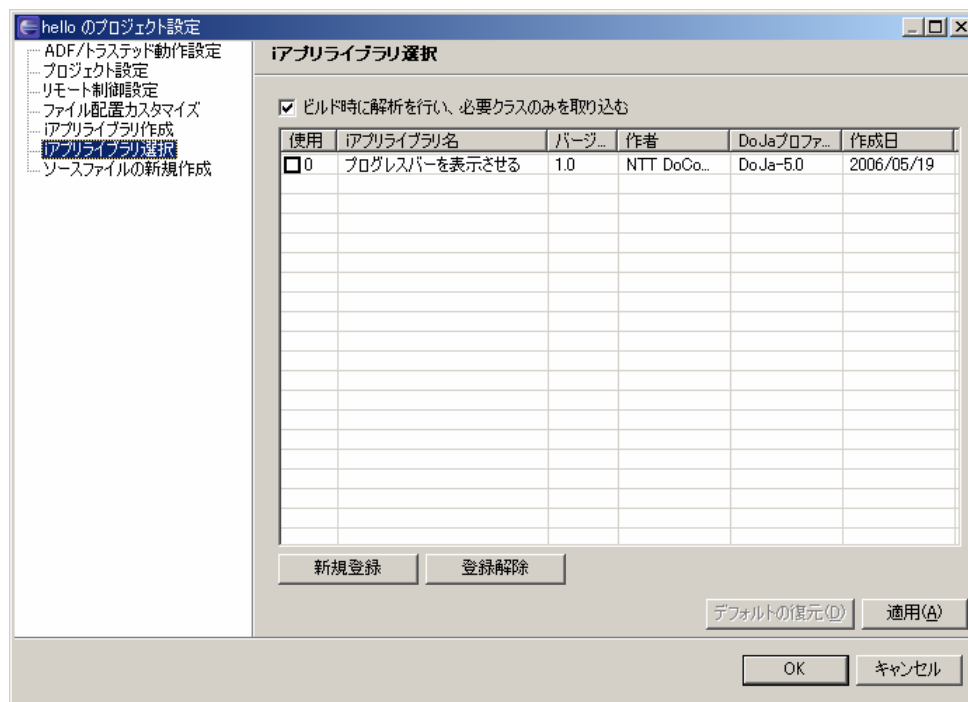


図 15-24 i アプリライブラリの選択画面 - ライブラリの追加



 ワンポイント

次の場所に i アプリライブラリを置いておくと自動的に登録できます。

C:\iDKDoJa5.0\lib\component

5. 不要になった i アプリライブラリは、一覧から選択して「登録解除」を押します。
(※ C:\iDKDoJa5.0\lib\component のライブラリは解除できません。)
6. 登録されたライブラリから使いたいライブラリだけにチェックをつけます。

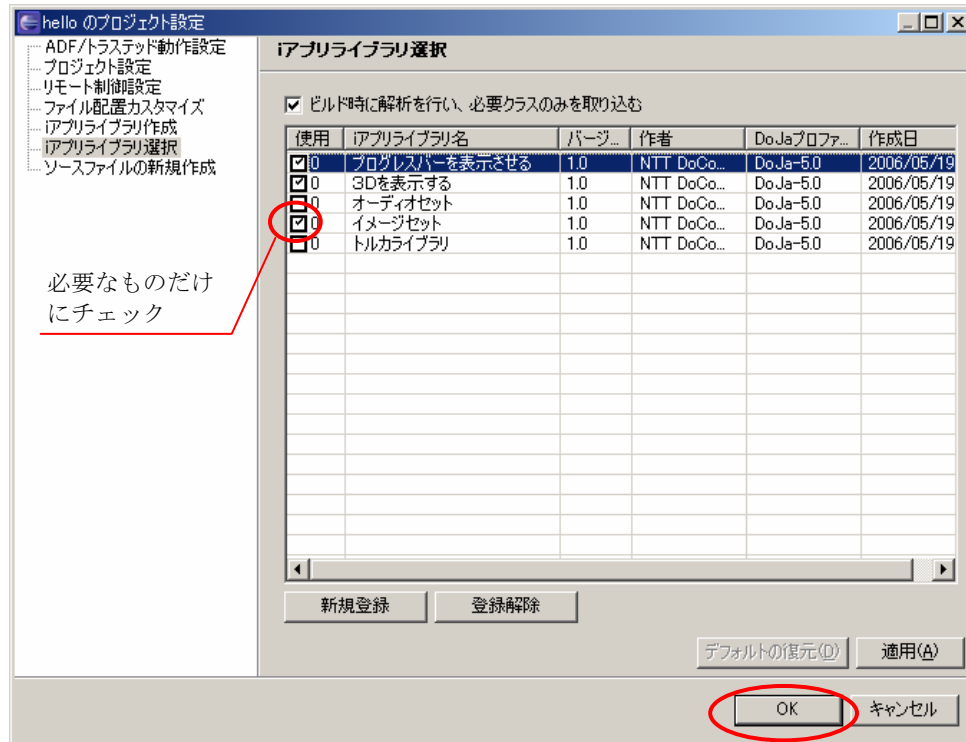


図 1 5-2 5 ライブラリの選択

7. 「OK」を押します。
チェックのついた i アプリライブラリが自動的に組み込まれ、i アプリからライブラリ中のクラスやメソッドを呼び出すことができますようになります。
この時、「ビルド時に解析を行い、必要クラスのみを取り込む」にチェックを付けていると、ライブラリ中で i アプリが使っていないクラスは、i アプリの Jar ファイルに取り込まれないようになります。
(全て取り込むと出来上がった Jar ファイルのサイズが膨大になるため。)
8. i アプリを実行します。

詳細については、「6.7.2. i アプリライブラリの利用」を参照してください。

15.3.14. デバッグの実行

アプリケーションをデバッグするには、次の手順に従います。^[11]

1. パッケージ・エクスプローラーからソース（HelloWorld.java）を開きます。
ソースに変数を定義して、変数の内容を表示させるとします。
たとえば、
 int a = 1;
と
 System.out.println("a = " + a);
を追加します。

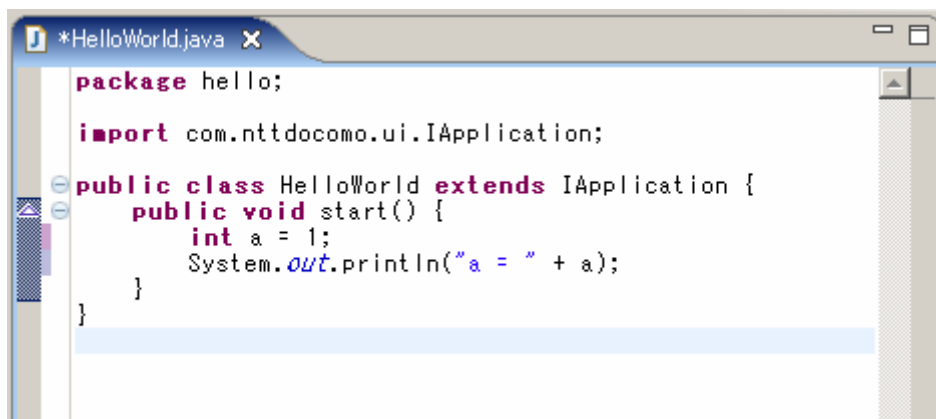


図 1 5-2 6 変数の表示

2. デバッグで止めたい行（ブレークポイント）の左枠をクリックしてチェックを付けます（図 1 5-2 7）。

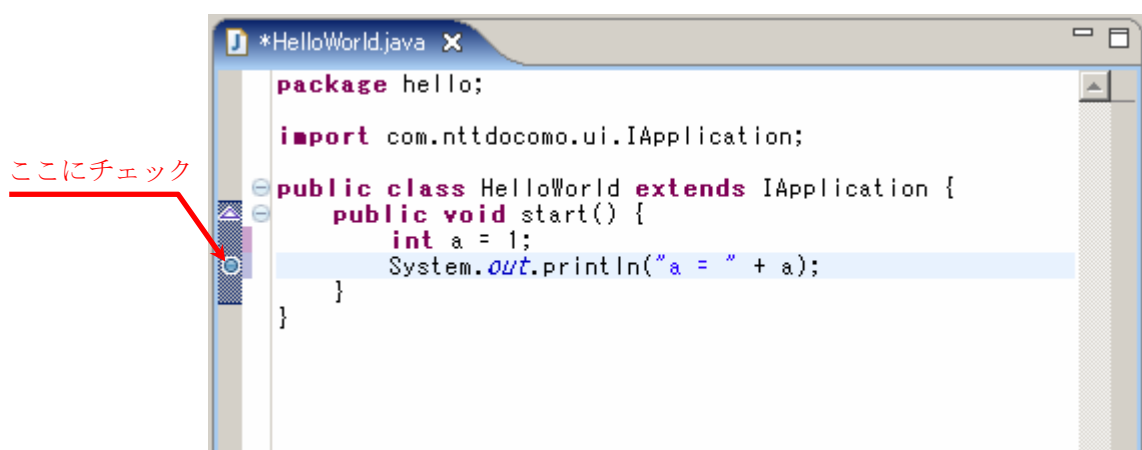


図 1 5-2 7 ブレークポイントの設定

^[11] エミュレータには、米国 Sun Microsystems 社が提供する CLDC の KDWP DEBUG PROXY (DEBUG AGENT) TOOL が組み込まれています。

3. 「実行」→「**構成および実行**」を押します。
4. 起動構成ウィンドウが表示されるので、起動構成の「**デバッグモード起動**」に**チェック**を入れて、「実行」を押します。

もし、デバッグ操作を無効にする場合は、実行画面の「デバッグモード起動」のチェックを外して下さい。

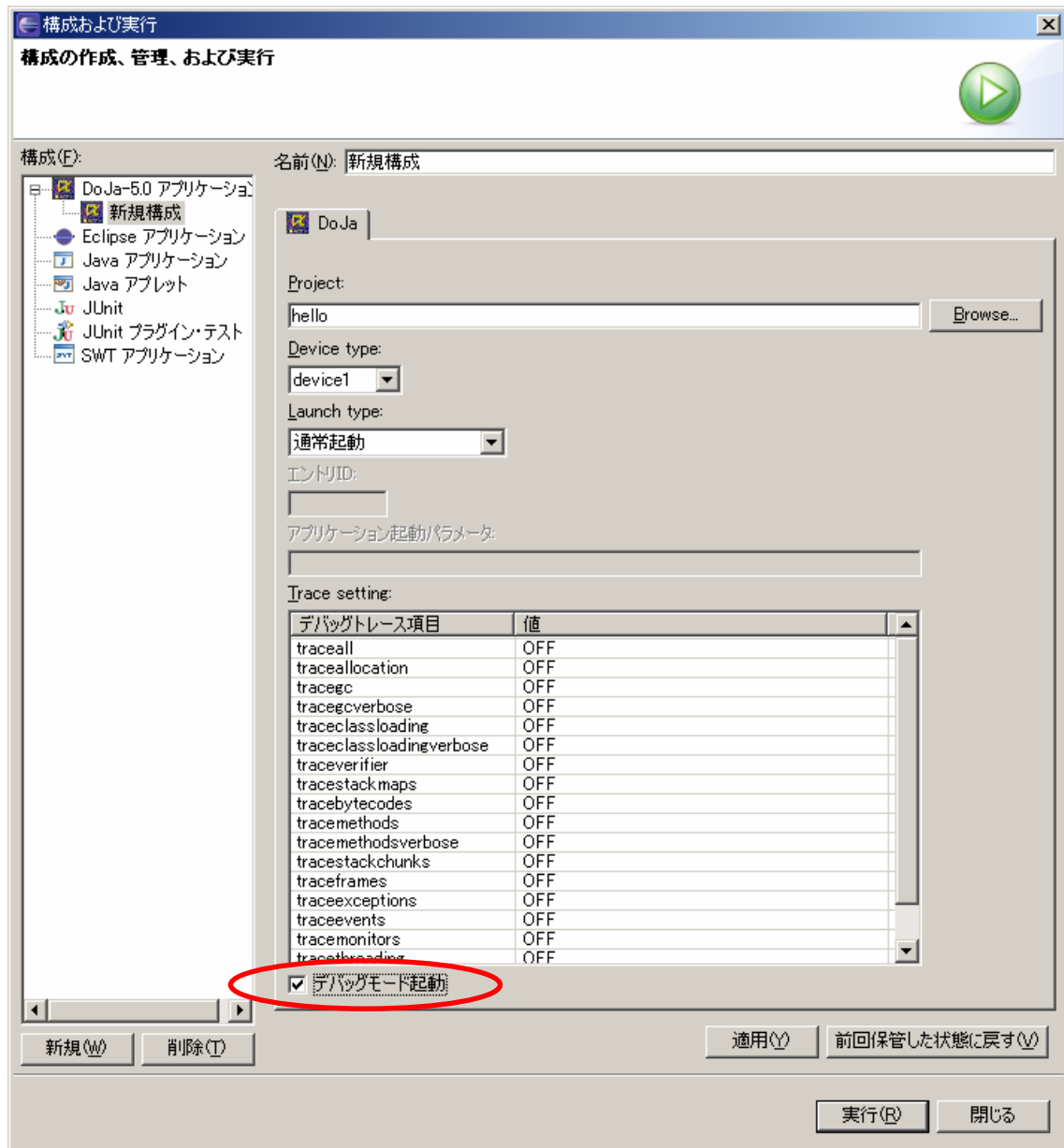


図 1 5-2 8 デバッグモードでの実行

5. 端末ウィンドウが表示されて、実行が止まったままになります（図 1 5-2 9）。この状態でエミュレータはデバッガ命令を待ち受けています。

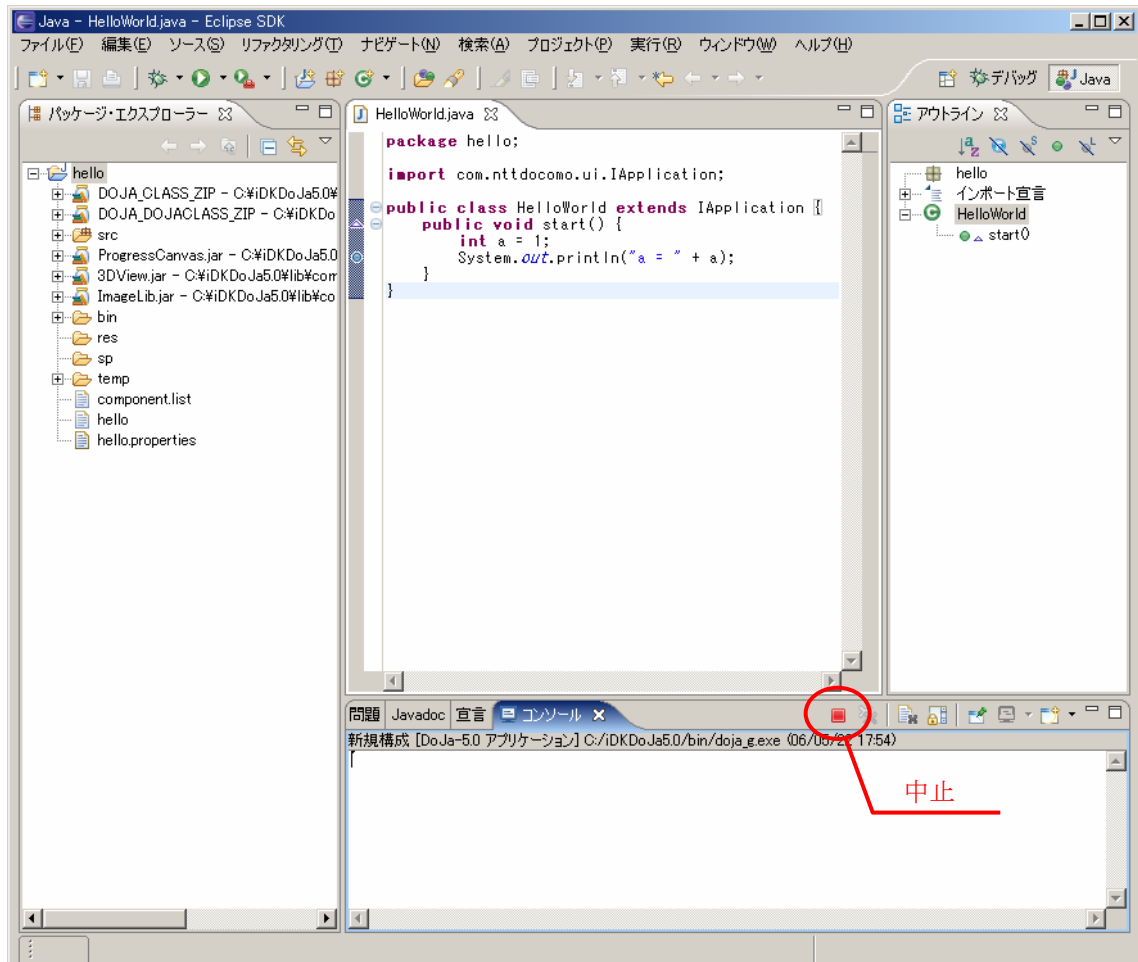


図 1 5-2 9 デバッグ実行 待ち受け状態

6. 次に「実行」 → 「構成およびデバッグ」を選択します。
7. デバッグ起動構成ウィンドウが表示されるので、構成の「リモート Java アプリケーション」をマウスでダブルクリック、もしくは、「新規」ボタンを押します。
8. ここでは「HelloWorld」が追加されるので、これを選択します。
「HelloWorld」の名前は、必要に応じて変更することができます。
9. 「接続プロパティ」に、
ホストを 「localhost」
ポートを 「8000」
に設定し「デバッグ」を押します（図 15-30）。
（この値は、一度設定するとそのまま保持されます。）

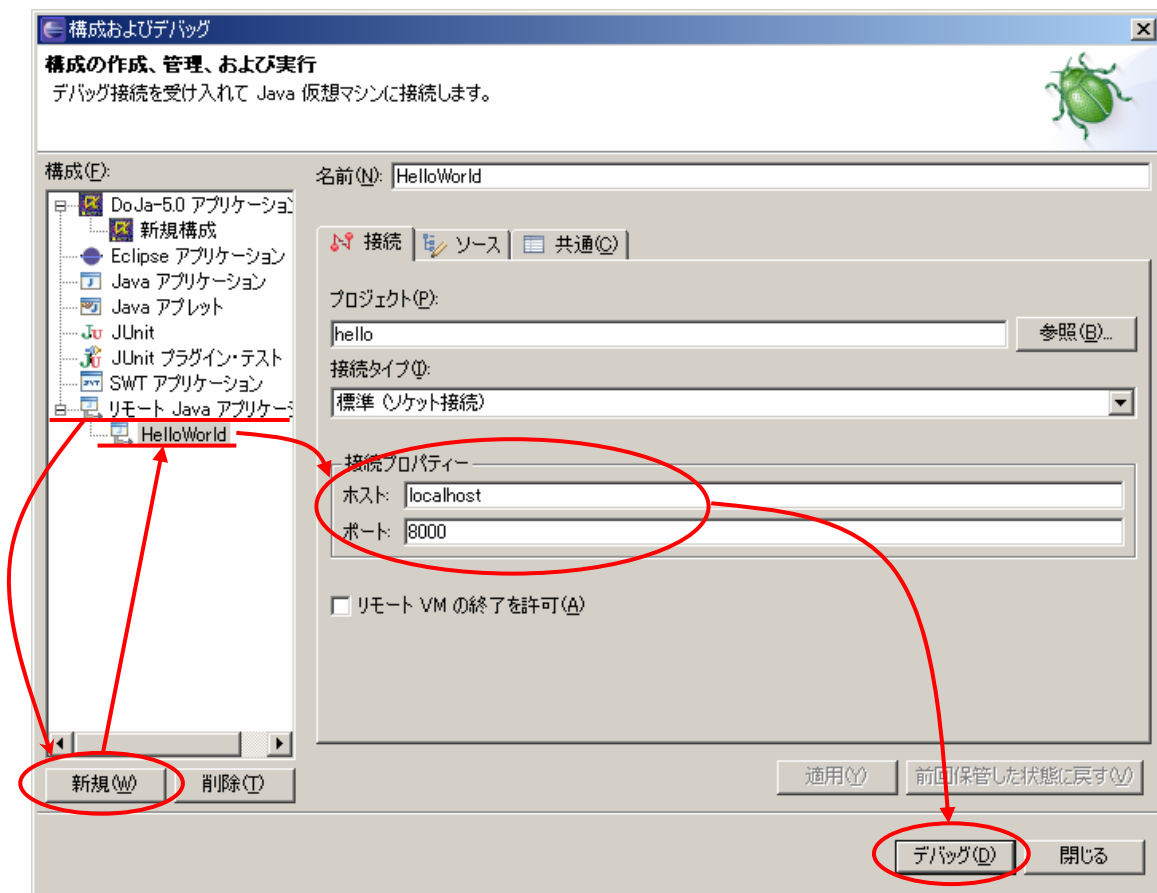


図 15-30 デバッグ接続画面

10. デバッグパースペクティブと端末ウィンドウが表示され、ブレークポイントで処理が中断します（図 1 5-3 1）。

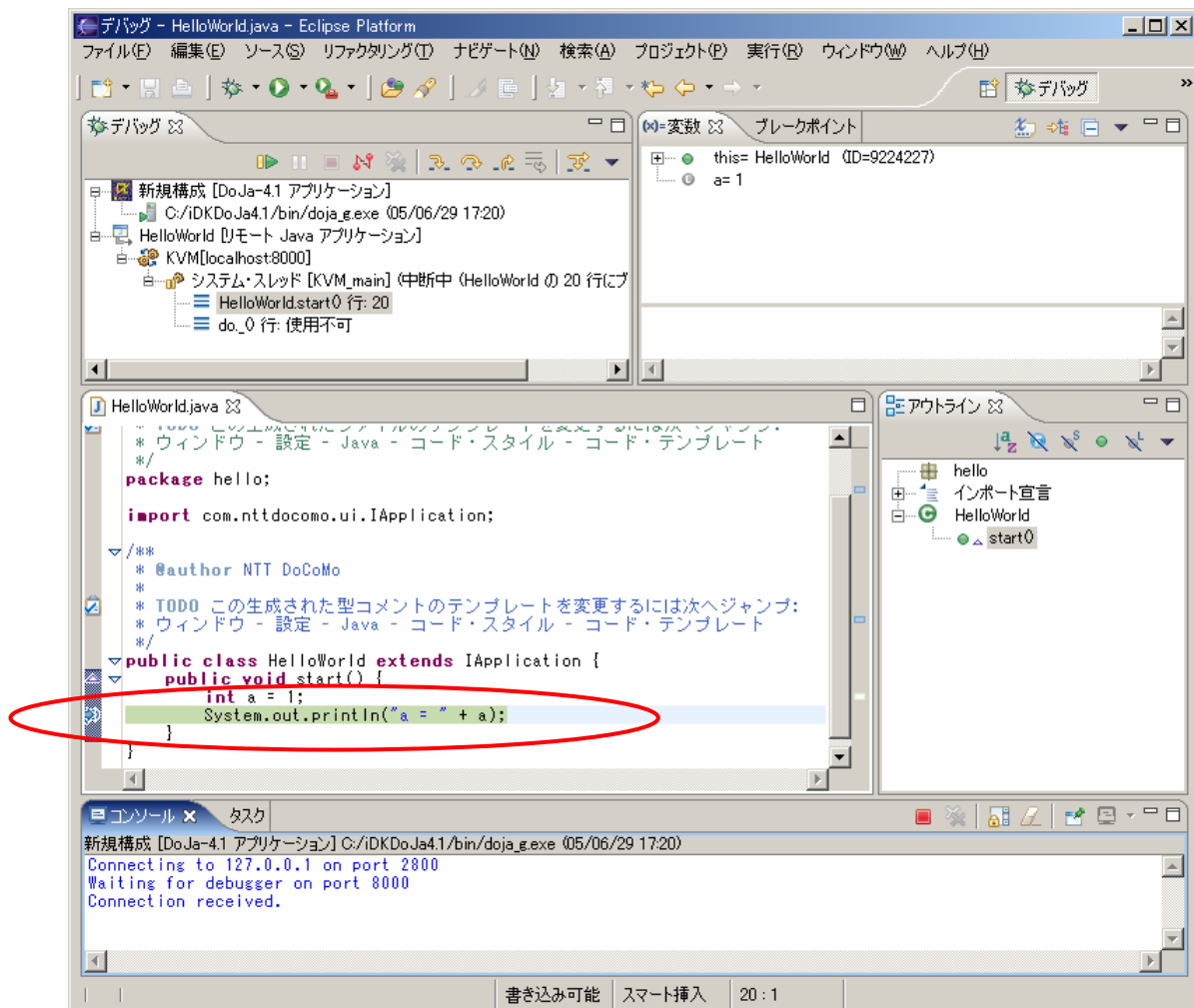


図 1 5-3 1 ブレークポイントでの処理の中断

11. 変数ウィンドウのウォッチポイントで、「a=1」を右クリックし、「値の変更」(もしくは「変数の変更」)を選択します(図 1 5-3 2)。

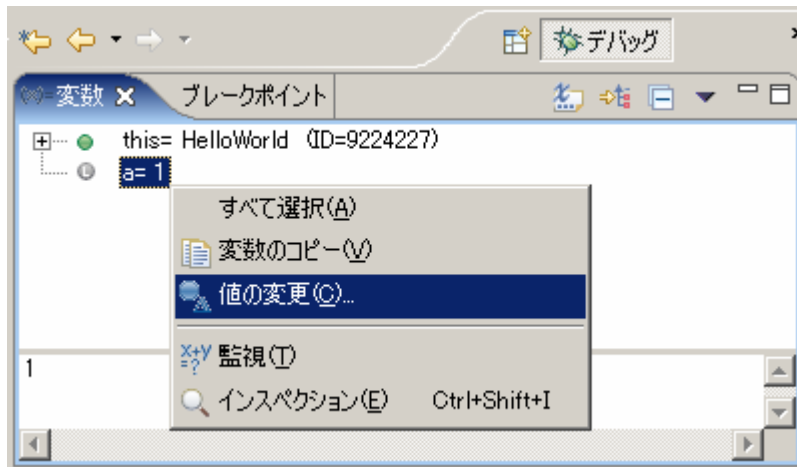


図 1 5-3 2 変数の変更

12. 「変数値を設定」ダイアログが表示されるので、変数の値を変更します。たとえば、aの値を10に変更します(図 1 5-3 3)。

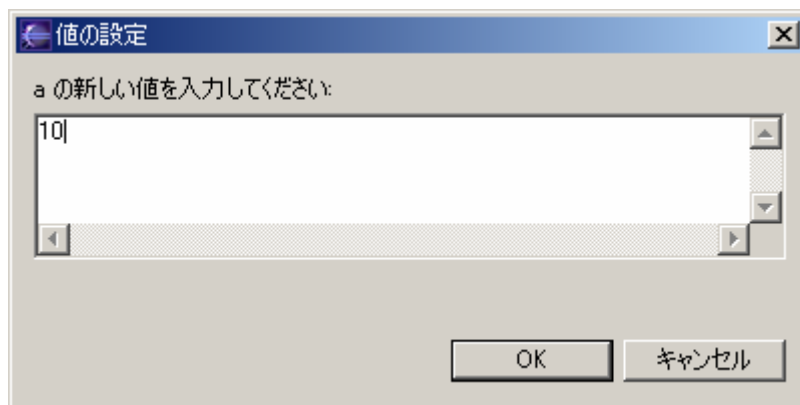


図 1 5-3 3 変数値の設定

13. 「デバッグビュー」 → 「再開」で中断した位置から処理を再開させます。
変数に変更された値で処理が再開されます。
「実行」メニューからも操作する事が可能です。
デバッグの実行は、デバッグビューでも様々なパターンをテストできます。

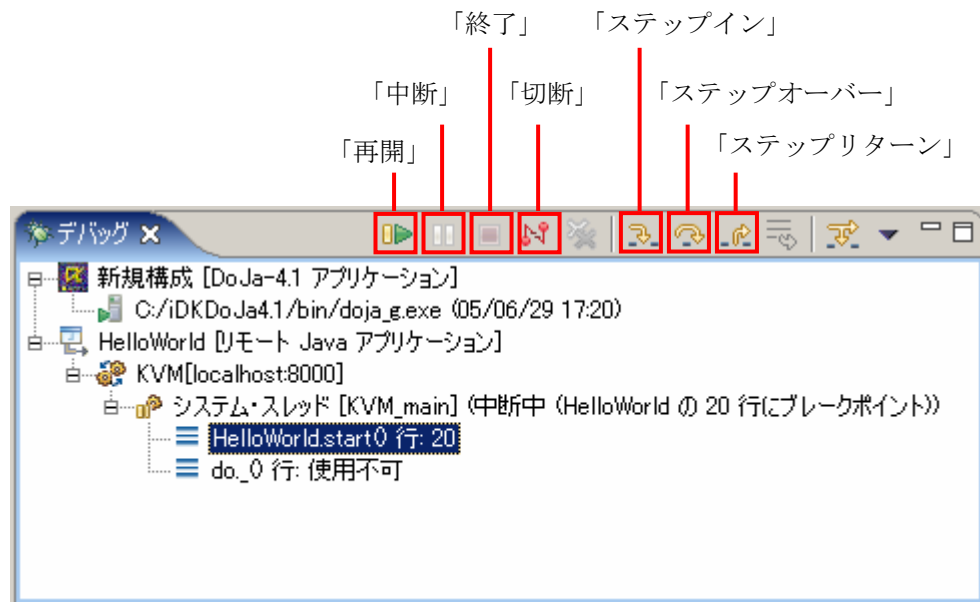


図 1 5-3 4 デバッグビューのボタン

14. 実行結果がコンソールウィンドウに a = 10 と表示されます。
もし、表示されない場合は、別のコンソールウィンドウに切り替えてください。

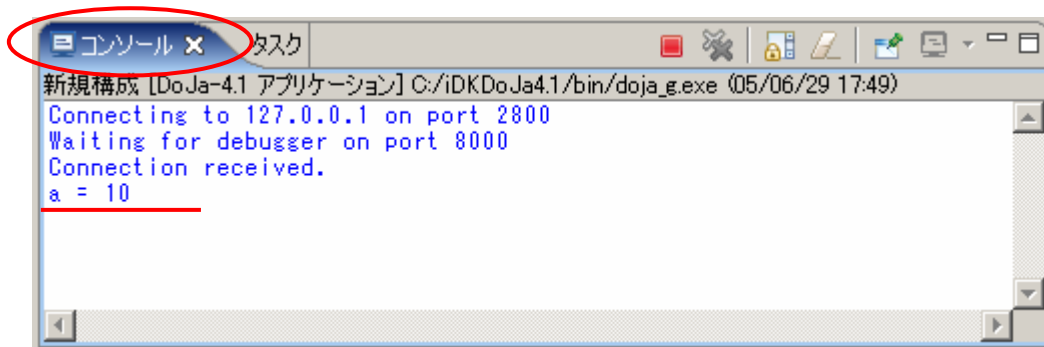


図 1 5-3 5 デバッグ実行結果

15. 端末ウィンドウの終話ボタンを押すか、「デバッグビュー」 → 「切断」でデバッグを終了します。

15.3.15. iアプリからDTV起動を行う場合の注意事項

Eclipseプラグイン上ではDTV画面（図 4-7 6）が存在しないため、Eclipseから起動したiアプリ上でDTV起動を行う場合は以下のような流れとなります（図 1 5-3 6）。

1. iアプリ上でDTV起動（"IApplication#launch(LAUNCH_DTV)"）を実行します（①）。
2. 起動確認画面で「はい」を選択するとiアプリが一時停止（サスペンド）し、エミュレータ上にDTV（ワンセグ）起動中のメッセージが表示されます（②）。
3. 端末ウィンドウの終話ボタンを押下する事でiアプリに復帰（レジューム）します（③）。

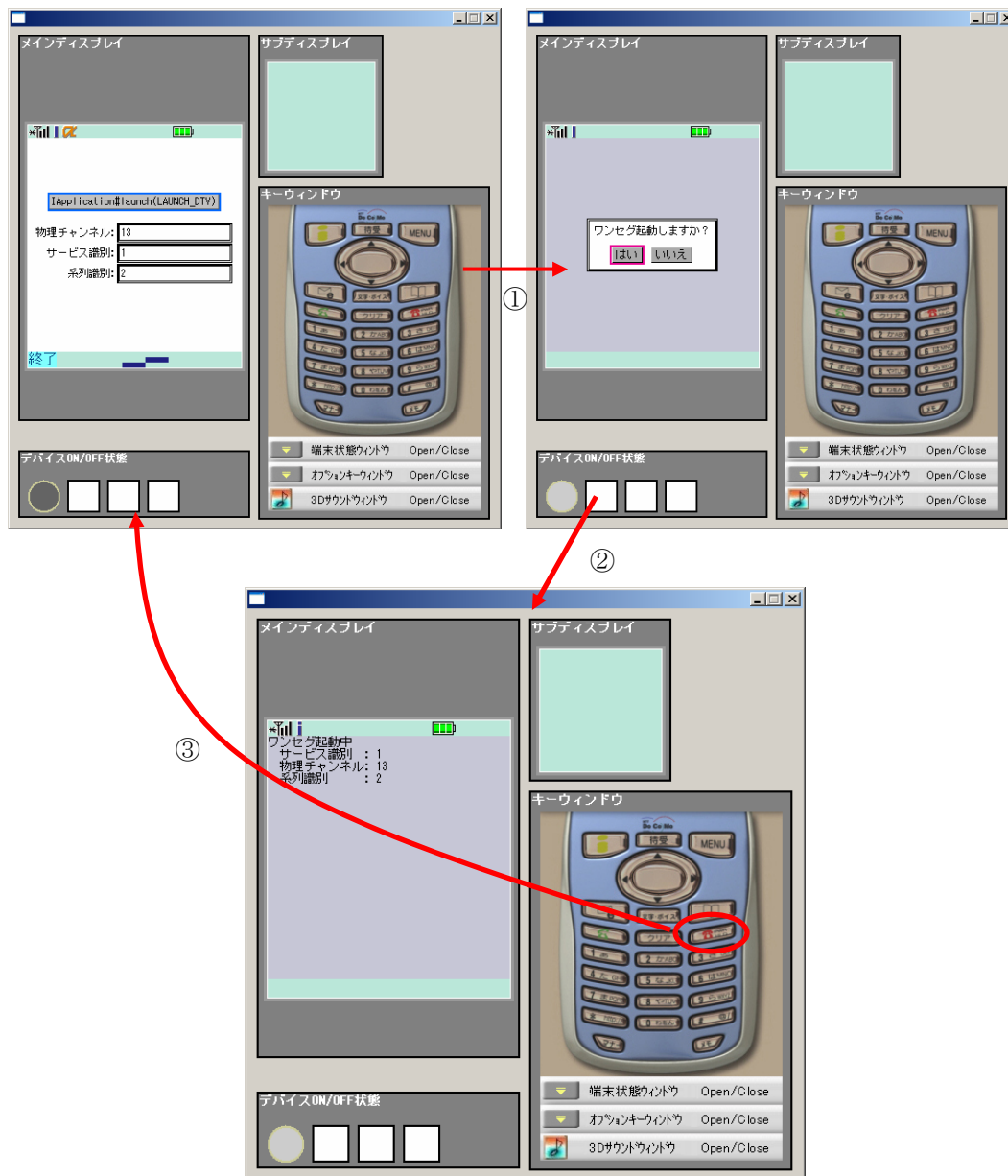


図 1 5-3 6 Eclipse から起動した i アプリ上でのDTV起動

15.4. エミュレータの設定

15.4.1. インストール先の設定

エミュレータのインストール先を設定することができます。
設定をするには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」を選択します（図 15-37）。

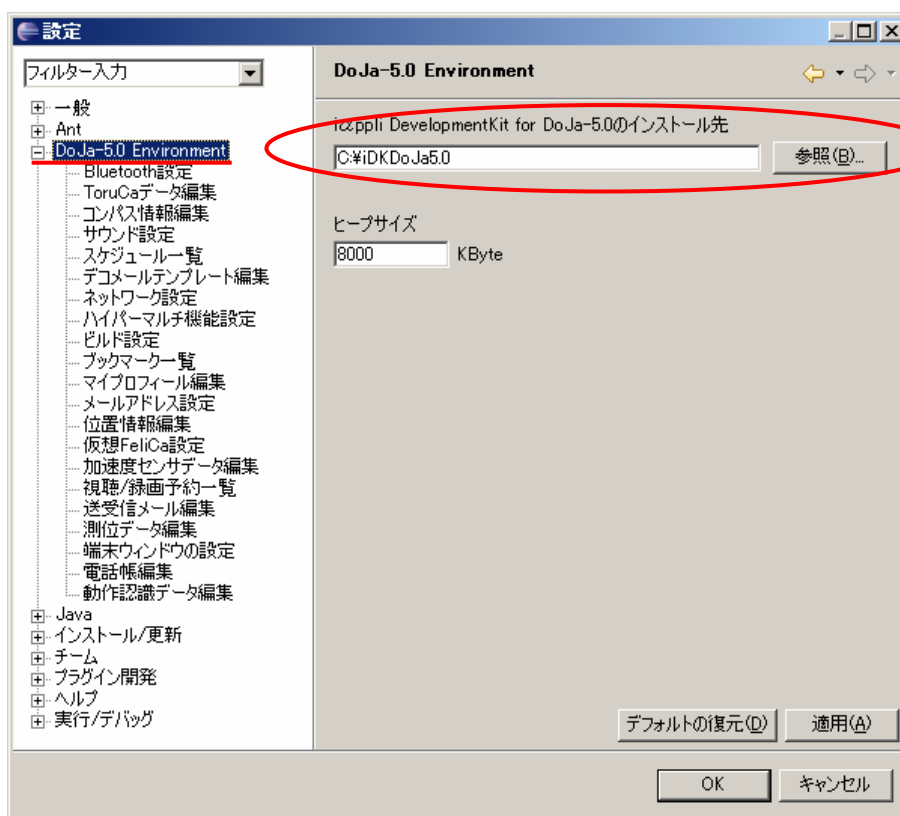


図 15-37 Eclipse 設定ウィンドウ

3. インストール先を設定して「OK」を押します。

15.4.2. ヒープサイズの設定

エミュレータのヒープサイズを設定することができます。設定をするには、次の手順に従ってください。

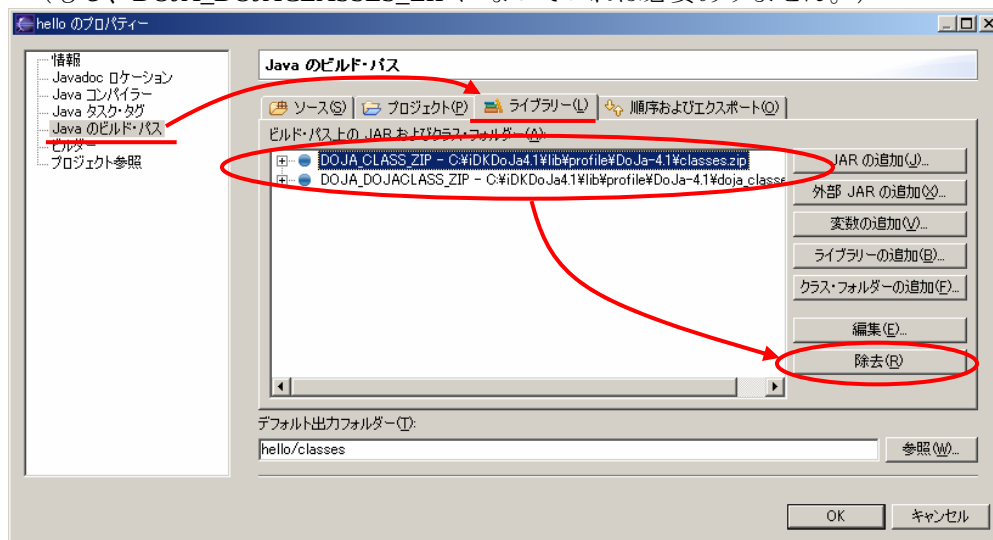
1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」を選択します（図 15-37）。
3. ヒープサイズを設定して「OK」を押します。

15.4.3. ターゲットプロファイルの変更

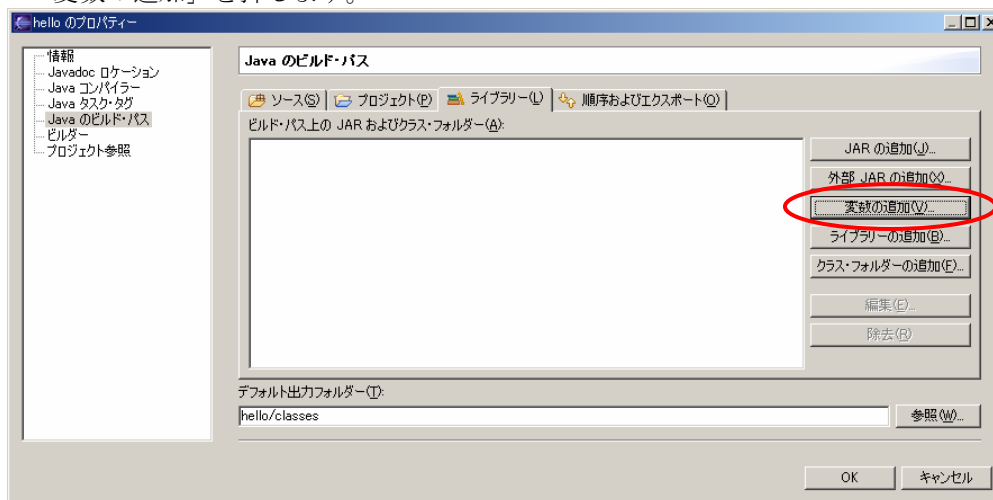
i アプリをビルドするターゲットプロファイルを変更します。
ターゲットプロファイルを変更するには、次の手順に従ってください。

1. 旧バージョンのプラグインで作成されたプロジェクトの場合は、以下の手順を実施してください。（プロジェクト毎に初回のみです。）
新規プロジェクトの場合は、手順2から実施してください。

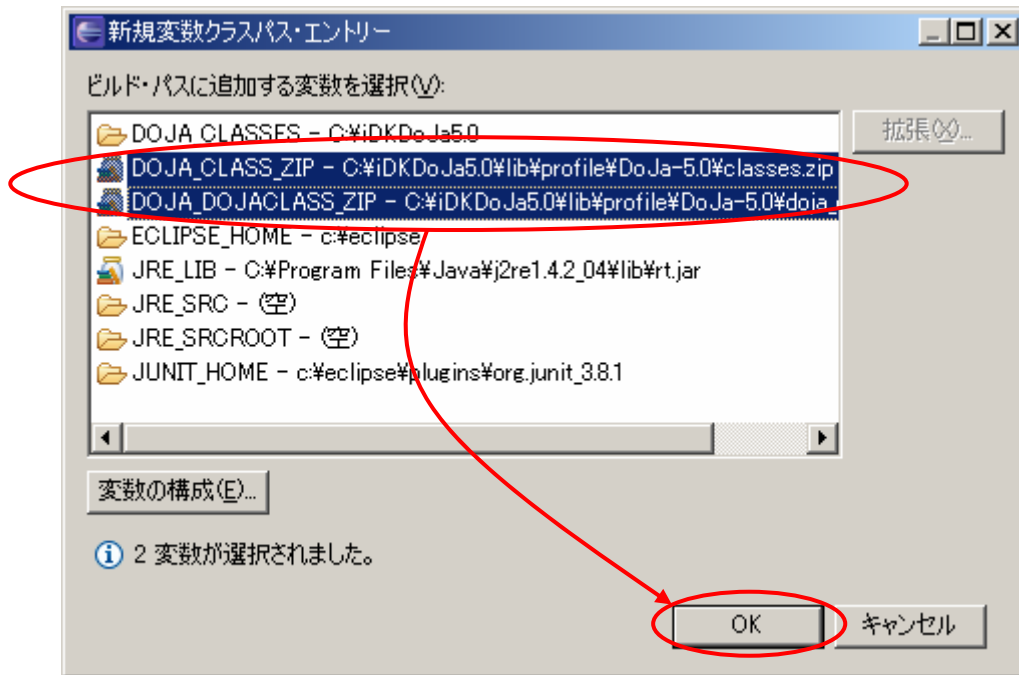
- ① 「プロジェクト」→「プロパティ」→「Java のビルド・パス」→「ライブラリ」を選択します。
- ② 「DOJA_CLASSES/lib/classes.zip」ライブラリを選択して「除去」を押します。
（もし、DOJA_CLASS_ZIP になっていれば必要ありません。）
- ③ 同様に「DOJA_CLASSES/lib/doja_classes.zip」ライブラリを選択して「除去」を押します。
（もし、DOJA_DOJACLASSES_ZIP になっていれば必要ありません。）



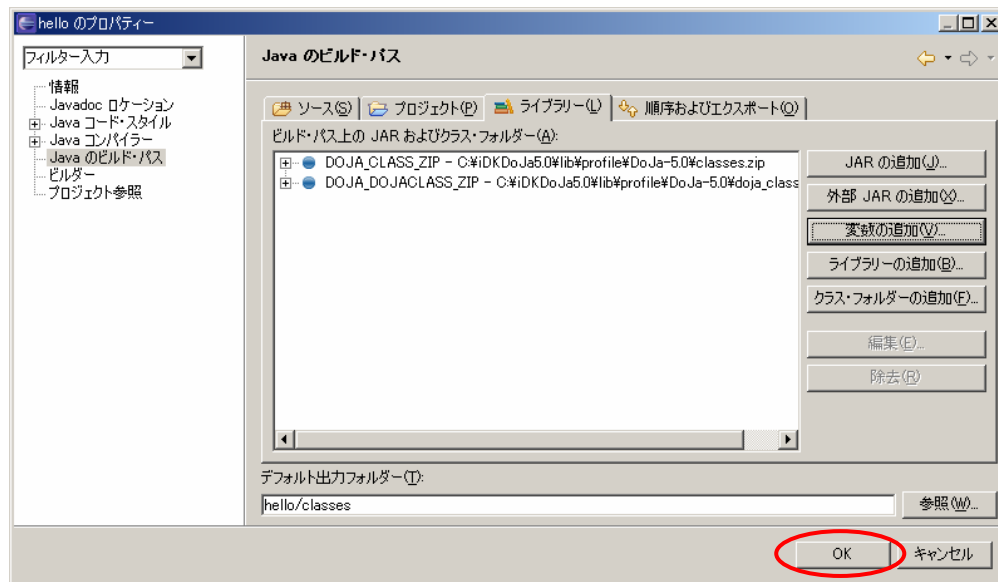
- ④ 「変数の追加」を押します。



- ⑤ 変数一覧から「DOJA_CLASS_ZIP」と「DOJA_DOJACCLASS_ZIP」を選択して「OK」を押します。



- ⑥ 一覧に選択した変数（「DOJA_CLASS_ZIP」と「DOJA_DOJACCLASS_ZIP」）が追加されていることを確認して「OK」を押します。



2. 「ウィンドウ」→「設定」を選択します。
3. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「ビルド設定」を選択します（図 1 5-3 8）。

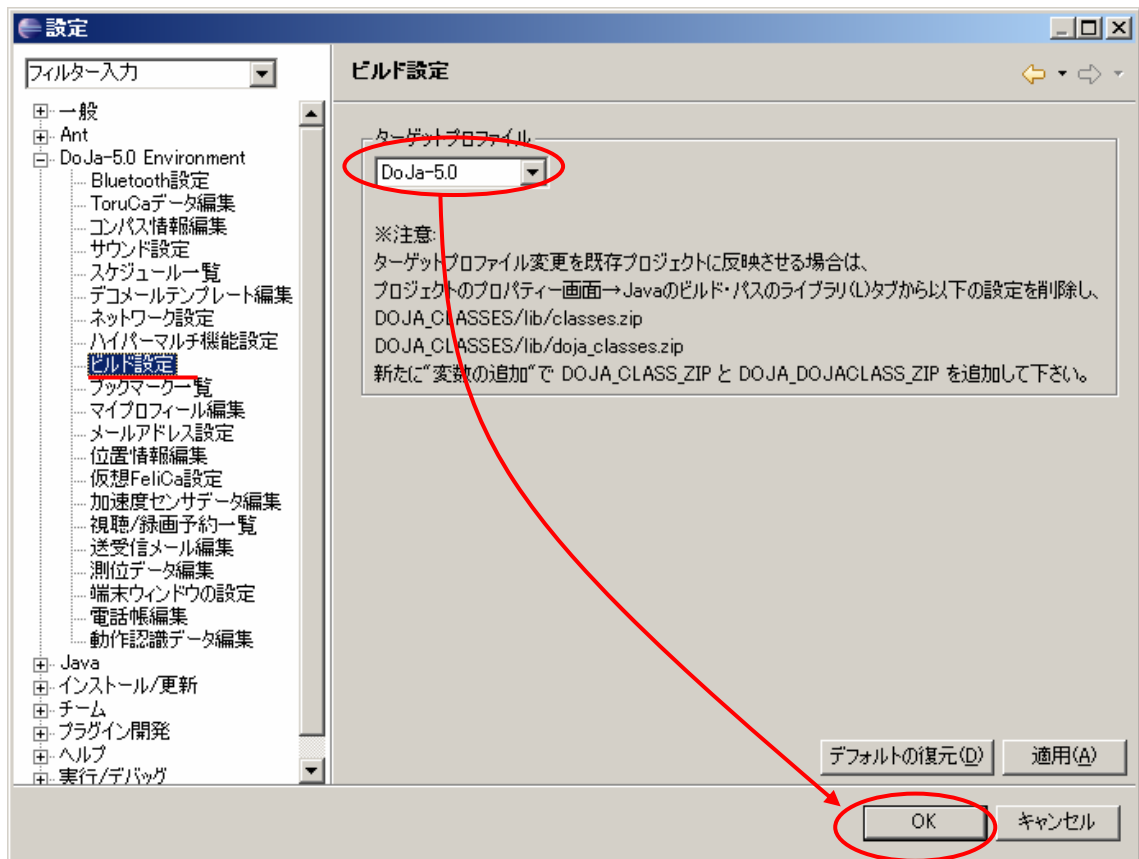


図 1 5-3 8 ビルド設定

4. ビルド対象にするターゲットプロファイルを選択して「OK」を押します。

※ ここで設定したプロファイルは、i アプリのビルド時に使われており、i アプリの動作には影響しません。

詳細については、「5.2 6.2. ターゲットプロファイルの設定」を参照してください。

15.4.4. ネットワークの設定

エミュレータのネットワークの設定を行うことができます。

設定をするには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「ネットワーク設定」を選択します（図 15-39）。

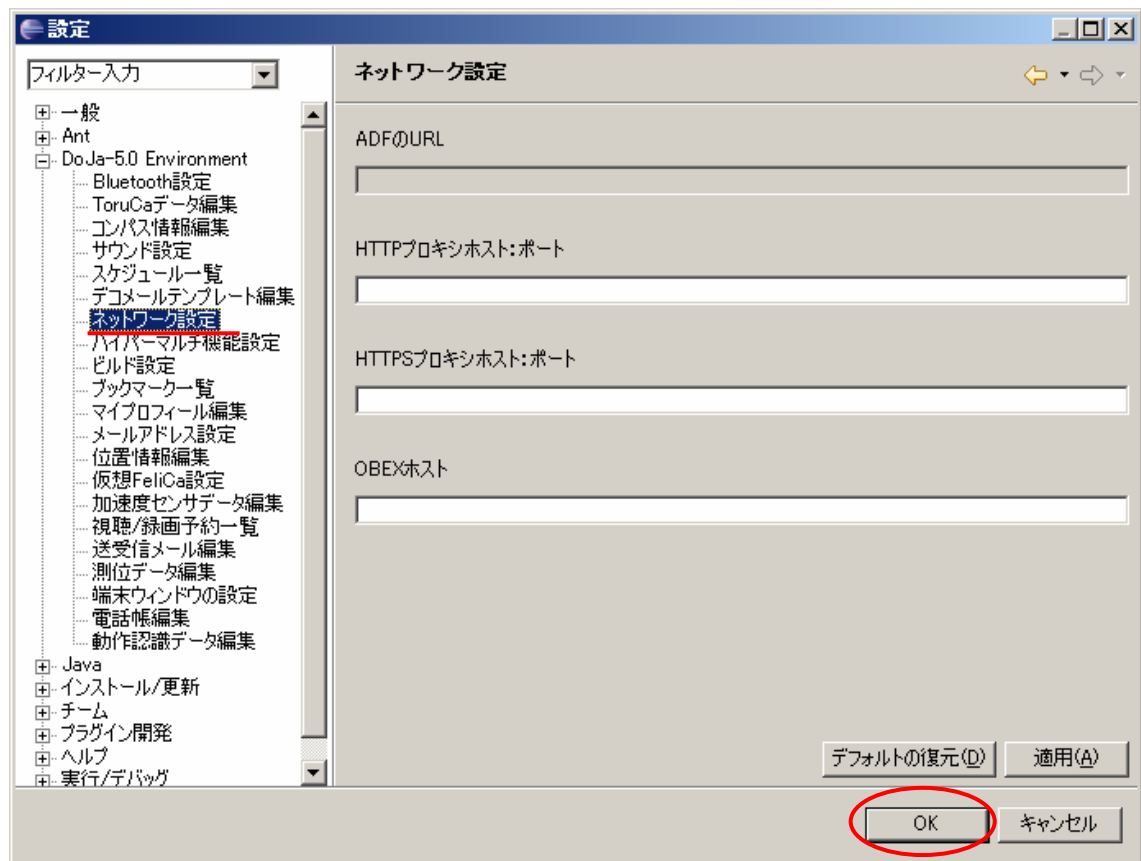


図 15-39 ネットワーク設定ウィンドウ

3. ネットワークの設定を変更して「OK」を押します。

ネットワークの各設定項目の詳細については、「5.1. ネットワークの設定」を参照してください。

15.4.5. メールアドレスの設定

エミュレータのメールアドレスの設定をすることができます。

設定をするには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「メールアドレス設定」を選択します（図 15-40）。

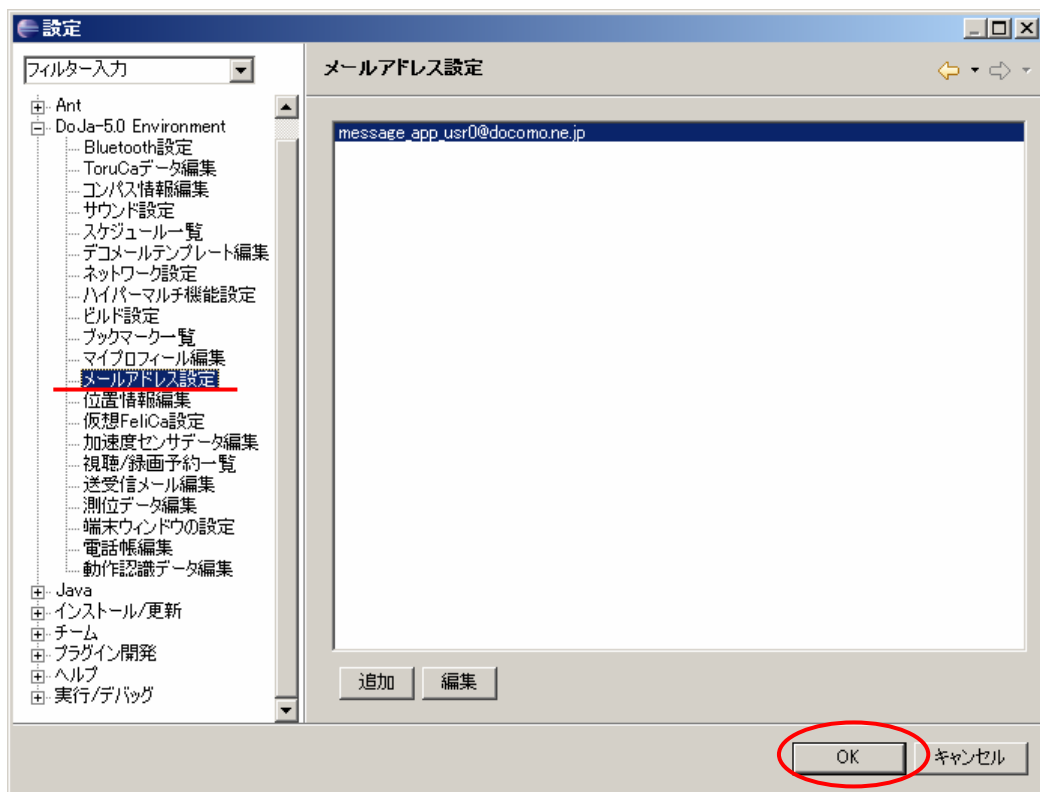


図 15-40 メールアドレス設定画面

3. メールアドレスを追加するには「追加」、変更するには変更したいアドレスを選択し、「編集」を押します。
メールアドレス変更画面が表示されます（図 15-41）。



図 15-41 メールアドレス変更画面

4. アドレスを変更後、「OK」を押します。
メールアドレス設定画面の一覧にアドレスの変更が反映されます。

15.4.6. 電話帳の編集

エミュレータの電話帳の編集をすることができます。
編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「電話帳編集」を選択します（図 15-42）。

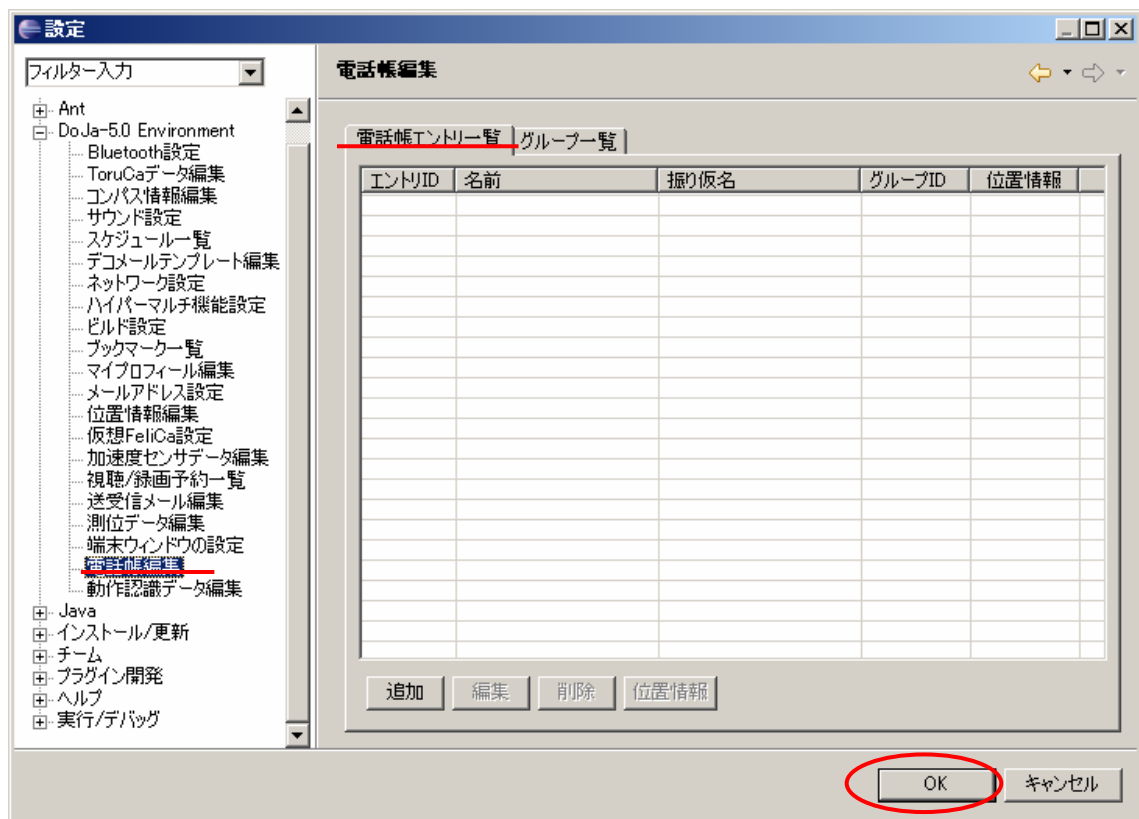


図 15-42 電話帳編集画面

3. 電話帳エントリの追加をするには「追加」、編集をするには「編集」を押します（図 1 5-4 3）。

電話帳エントリ登録

名前 姓 DoCoMo

名 太郎

振り仮名 姓 ドコモ

名 タロウ

電話番号 1 090XXXXXXXX

2 03XXXXXXXX

3

メールアドレス 1 message_app_usr1@

2

3

グループID 1

設定 取消

図 1 5-4 3 電話帳エントリ画面

4. 各設定項目を設定します。
5. 「設定」を押すと、電話帳一覧に追加されます。

各設定項目の詳細については、「表 5-7. 電話帳エントリの設定項目」を参照してください。

6. 電話帳エントリを削除するには、一覧から電話帳エントリを選択して「削除」を押します。
電話帳エントリファイルが削除され、電話帳一覧からエントリが削除されます。

7. 電話帳エントリに位置情報を追加するには、一覧から電話帳エントリを選択して「位置情報」を押します（図 1 5 - 4 4）。

図 1 5 - 4 4 電話帳位置情報編集画面

15.4.7. グループの編集

エミュレータのグループの編集をすることができます。
編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「電話帳編集」を選択し、グループ一覧タブをクリックします（図 15-45）。

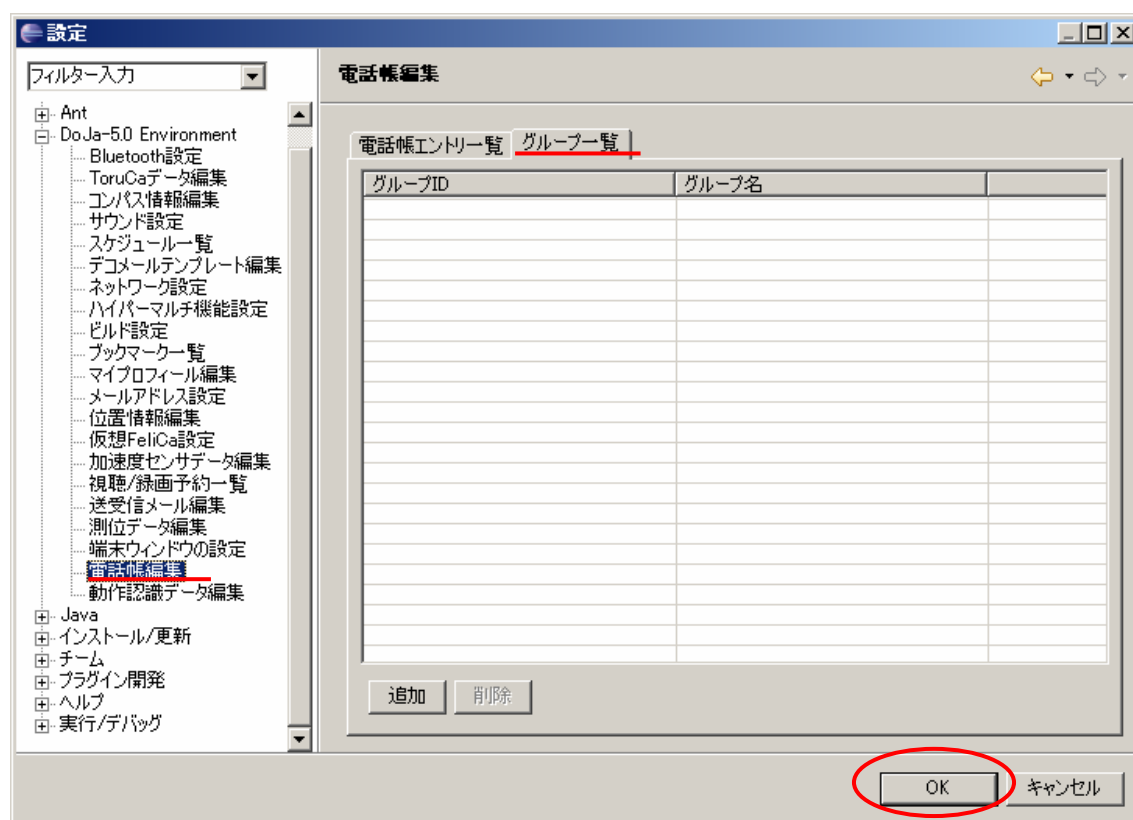


図 15-45 Eclipse 設定ウィンドウのグループ編集画面

3. グループを追加したい場合は「追加」を押します（図 15-46）。

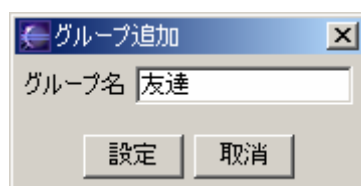


図 15-46 グループ追加画面

4. 「設定」を押します。
グループ一覧にグループ名が追加されます。
5. グループを削除したい場合は、一覧からグループを選択し、「削除」を押します。
グループファイルが削除され、一覧からグループ名が削除されます。

15.4.8. 送受信メールの編集

エミュレータの送受信メールの編集をすることができます。
編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「送受信メール編集」を選択します（図 15-47）。

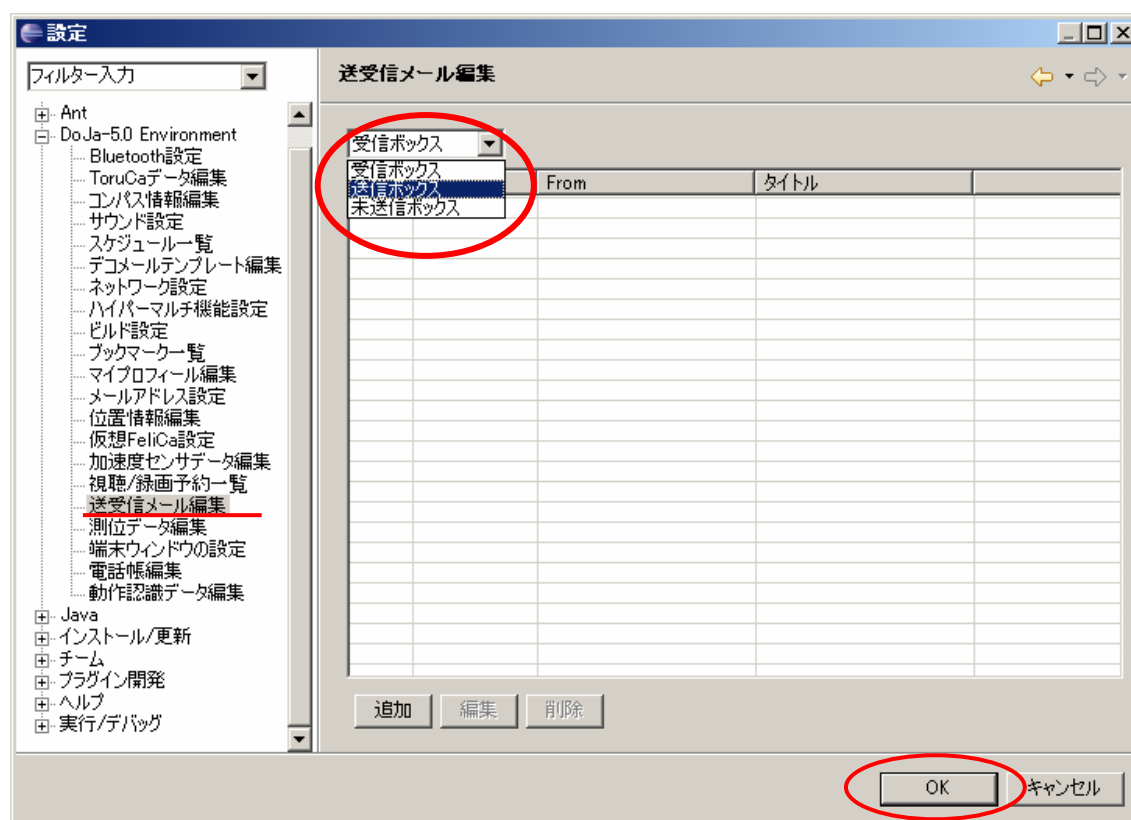


図 15-47 Eclipse 設定ウィンドウの送受信メール編集画面

3. 編集したいメールボックスを選択すると、一覧が切り替わります。
メールボックスの種類は、「受信ボックス」「送信ボックス」「未送信ボックス」の3種類があります。
4. メールを追加したい場合は「追加」、メールを編集したい場合は一覧から編集したいメールを選択して、「編集」ボタンを押します(図 15-48、図 15-49)。

メール編集(受信ボックス)

To: message_app_usr0@docomo.ne.jp

From: message_app_usr0@docomo.ne.jp

タイトル: おめでとう

受信日: 2002 年 12 月 01 日 10 時 11 分

MessageCode: 0000000000

ネイティブメール: ☐

状態: ☒ 既読 ☐ 未読

本文

お誕生日おめでとう。

設定 取消

図 1 5-4 8 受信メール編集画面

メール編集(送信ボックス)

To: message_app_usr0@docomo.ne.jp

From: message_app_usr0@docomo.ne.jp

タイトル: Re:おめでとう

送信日: 2002 年 12 月 01 日 10 時 22 分

MessageCode: 0000000000

ネイティブメール: ☐

宛先状態: ☒ XString ☐ 通常文字列

本文

ありがとう。

設定 取消

図 1 5-4 9 送信/未送信メール編集画面

5. 各設定項目を設定します。
6. 「OK」を押すと、送受信メール編集一覧に追加されます。

各設定項目の詳細については、「5.1 6. 送受信メールの編集」を参照してください。

15.4.9. スケジュール一覧

エミュレータのスケジュール一覧を編集することができます。

表示するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「スケジュール一覧」を選択します（図 15-50）。

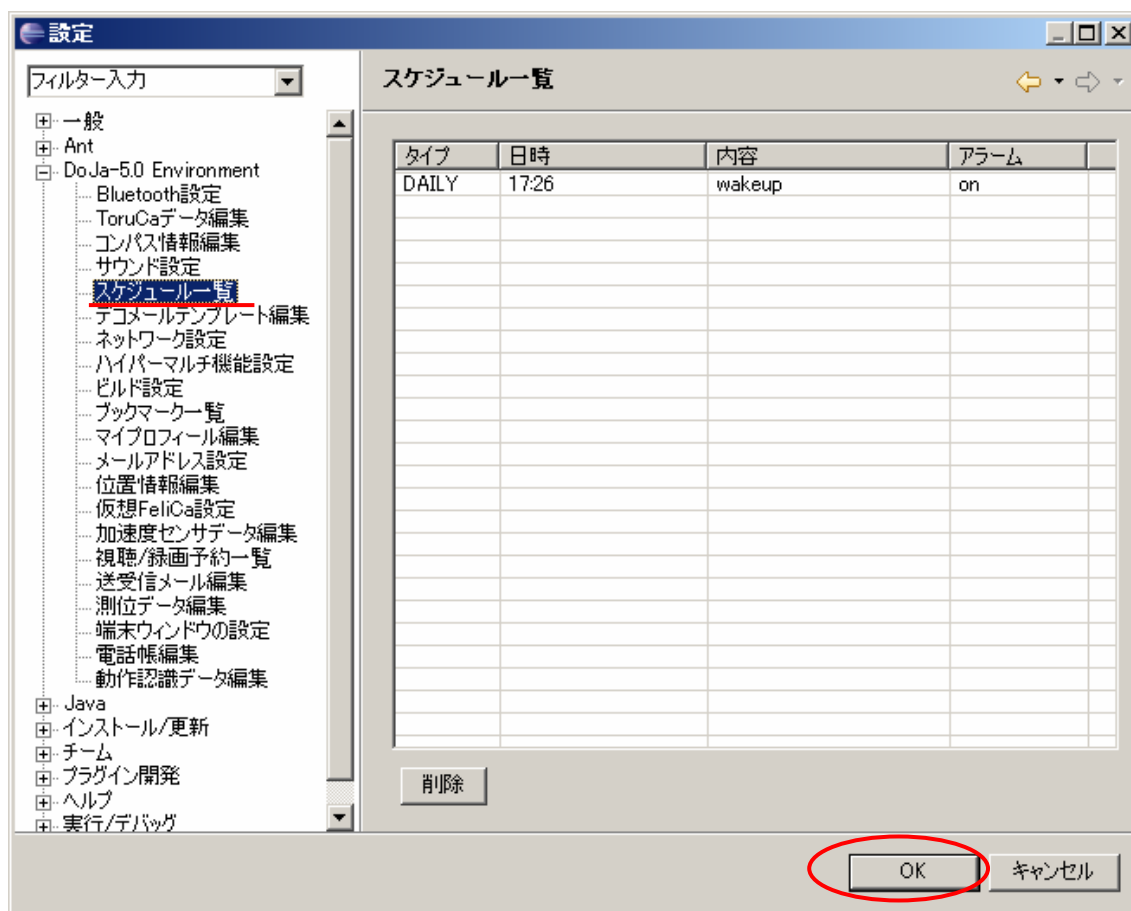


図 15-50 スケジュール一覧画面

3. スケジュールを削除するには、一覧からスケジュールを選択して「削除」を押します。スケジュールファイルが削除され、一覧からスケジュールが削除されます。

15.4.10. ブックマーク一覧

エミュレータのブックマーク一覧を編集することができます。
表示するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「ブックマーク一覧」を選択します（図 15-51）。

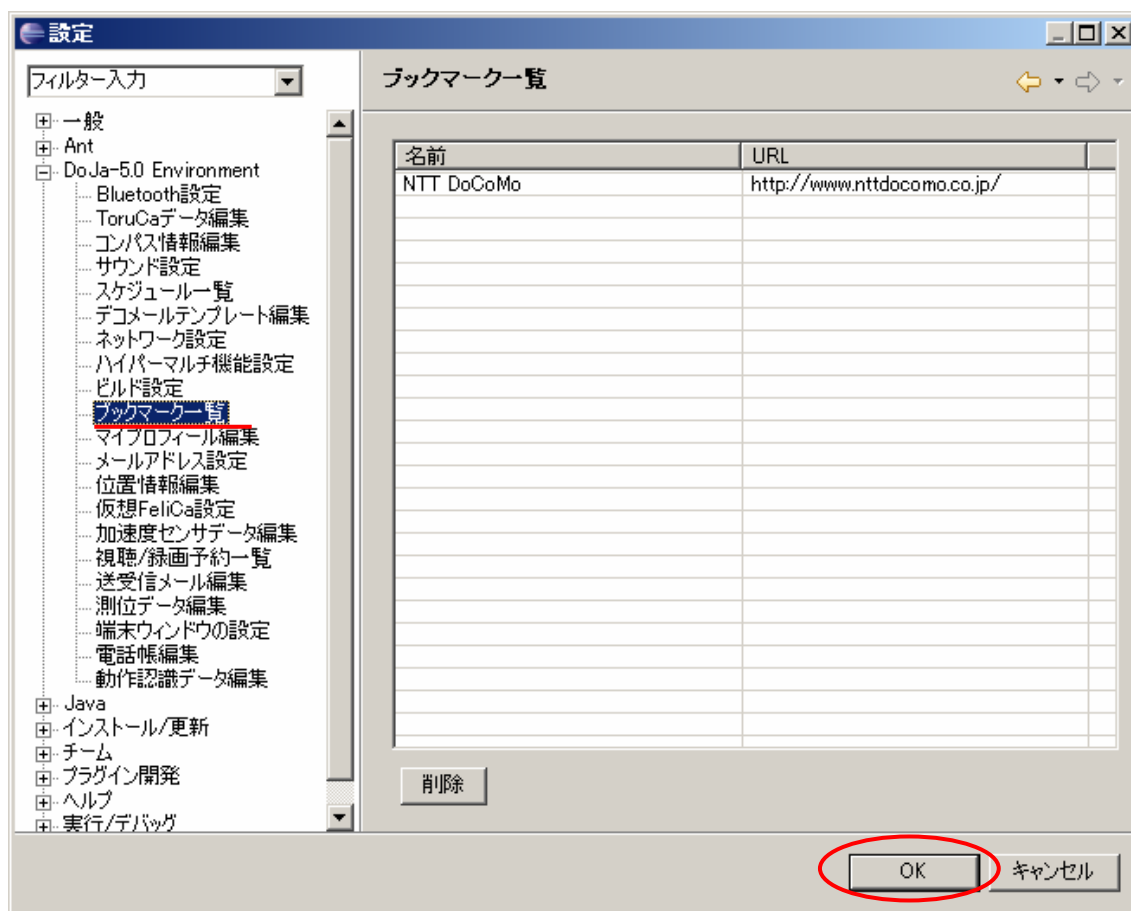


図 15-51 ブックマーク一覧画面

3. ブックマークを削除するには、一覧からブックマークを選択して「削除」を押します。ブックマークファイルが削除され、一覧からブックマークが削除されます。

15.4.11. サウンド設定

サウンドの再生モードと代替音を設定することができます。

設定をするには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「サウンド設定」を選択します（図 1 5-5 2）。

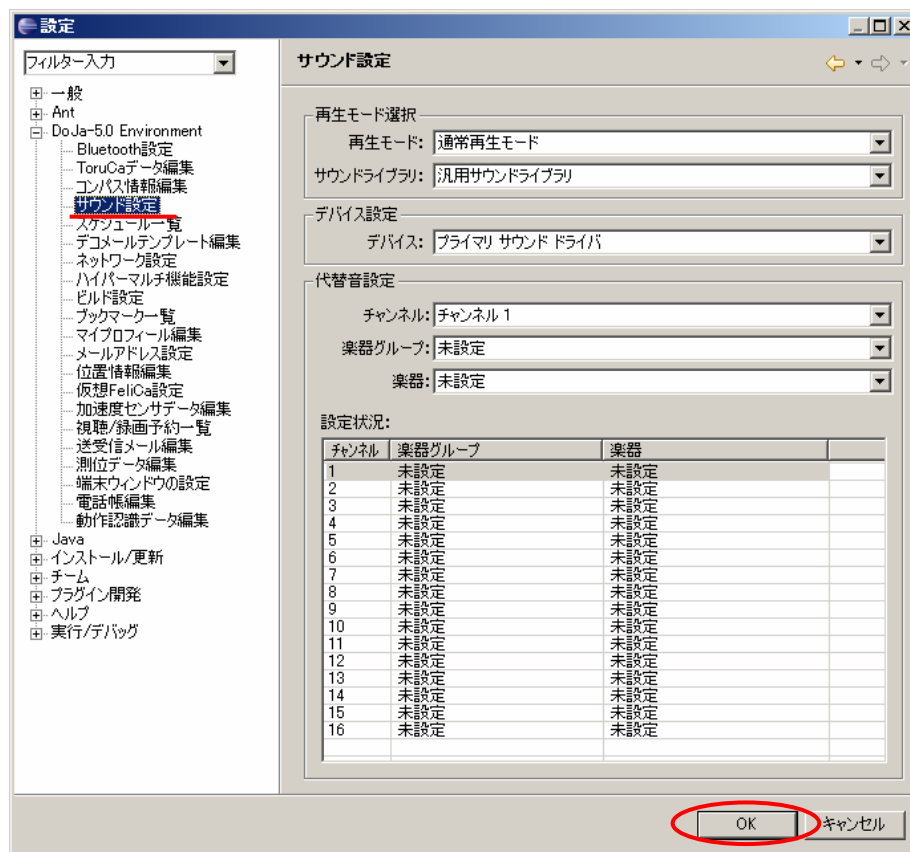


図 1 5-5 2. サウンド設定画面

3. 各設定項目を設定します。
サウンドライブラリの選択とデバイス設定は、再生モードが「通常再生モード」の場合に有効です。
代替音設定は、再生モードが「通常再生モード」、サウンドライブラリが「汎用サウンドライブラリ」の場合に有効です。
「通常再生モード」でのサウンド再生には、別途、**DirectX** が必要です。
4. 「OK」を押すと、設定されます。

各設定項目の詳細については、「5.6. サウンド設定」を参照してください。

15.4.12. 位置情報編集

エミュレータの位置情報を編集することができます。

編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「位置情報編集」を選択します（図 1 5-5 3）。

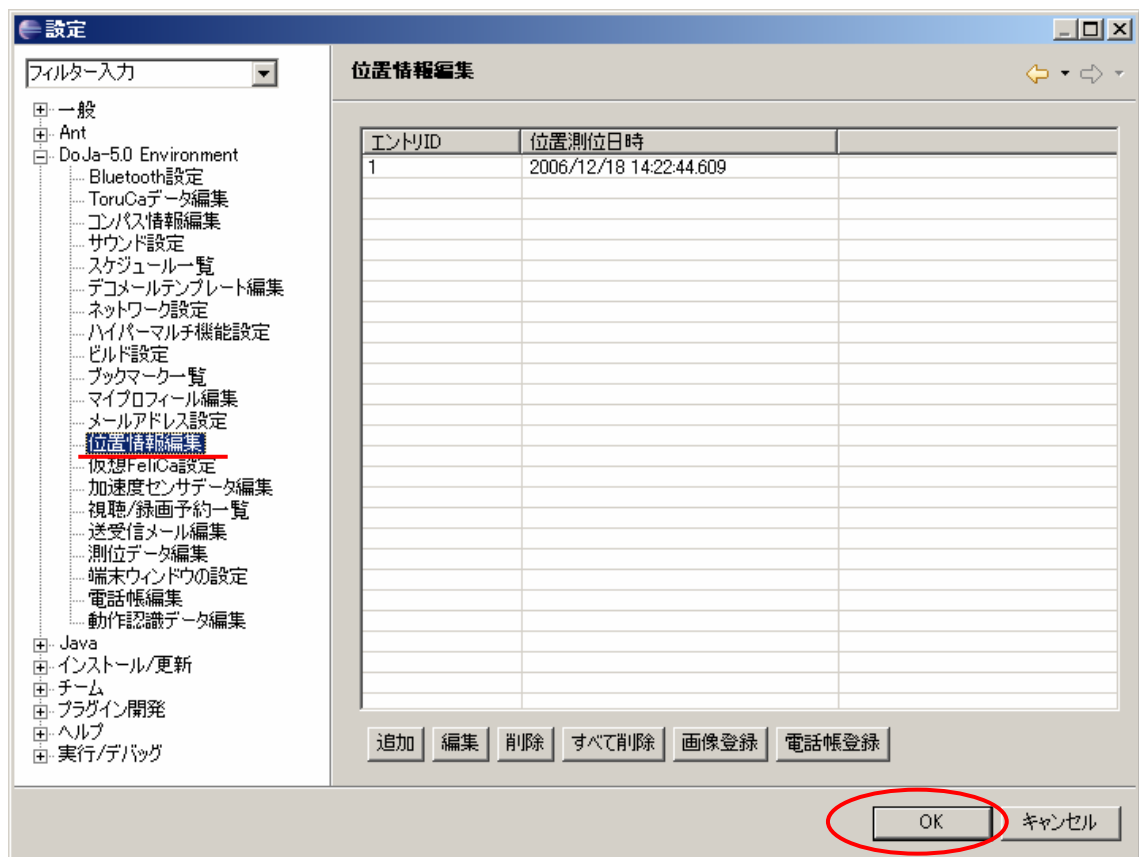


図 1 5-5 3 Eclipse 設定ウィンドウの位置情報編集一覧画面

3. 位置情報を追加したい場合は「追加」、位置情報を編集したい場合は一覧から編集したい位置情報を選択して、「編集」ボタンを押します（図 1 5-5 4）。

位置情報編集

エントリID 1

位置測位日時
2006 / 05 / 01 01 : 01 : 00 . 000

測位レベル
☐ 1 ☐ 2 ☒ 3

緯度
N 00 . 00 . 00 . 000

経度
E 000 . 00 . 00 . 000

高度
0 m

測地系 WGS84

設定 取消

図 1 5 - 5 4 位置情報編集画面

4. 各設定項目を設定します。
5. 「設定」を押すと、位置情報編集一覧に追加されます。
6. 位置情報を削除したい場合は、一覧から位置情報を選択し、「削除」を押します。
「すべて削除」を押すと位置情報が全て削除されます。
7. 画像に位置情報を登録したい場合は、一覧から位置情報を選択し、「画像登録」を押します。
エントリIDの入力画面が表示されるので、画像のエントリIDを入力して「設定」を押します。
8. 電話帳に位置情報を登録したい場合は、一覧から位置情報を選択し、「電話帳登録」を押します。
エントリIDの入力画面が表示されるので、電話帳編集で登録した電話帳のエントリIDを入力して「設定」を押します。

各設定項目の詳細については、「5.17. 位置情報編集」を参照してください。

15.4.13. 測位データ編集

エミュレータのGPS測位データ（測位結果）を編集することができます。

編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「測位データ編集」を選択します（図 15-55）。

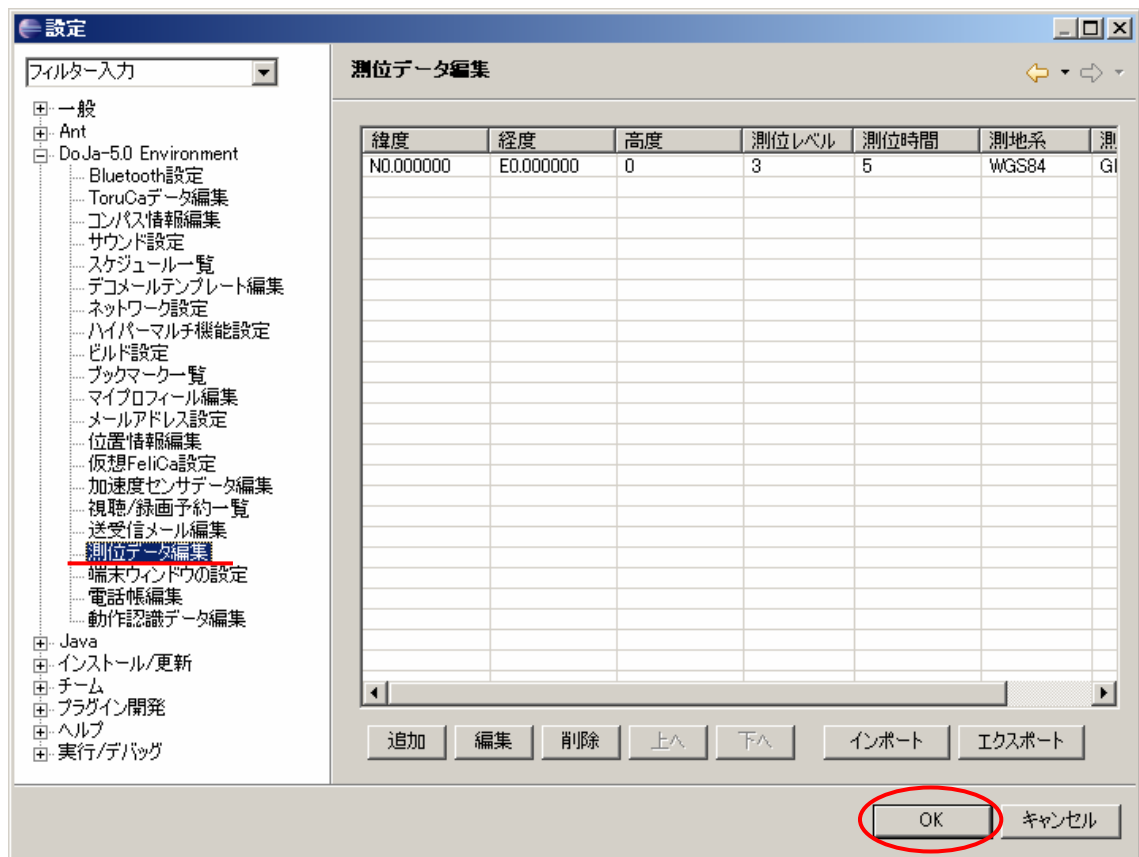


図 15-55 Eclipse 設定ウィンドウの測位データ編集一覧画面

3. 測位データを追加したい場合は「追加」、測位データを編集したい場合は一覧から編集したい測位データを選択して、「編集」ボタンを押します。

図 1 5-5 6 測位データ編集画面

4. 各設定項目を設定します。
5. 「設定」を押すと、測位データ一覧に追加されます。
6. 測位データの順序を変更したい場合は、順序を変更したい測位データを選択して「上へ」もしくは「下へ」を押します。
7. 測位データを削除したい場合は、一覧から測位データを選択し、「削除」を押します。

各設定項目の詳細については、「5.1 8. 測位データ編集」を参照してください。

15.4.14. コンパス情報編集

エミュレータの電子コンパスの情報を編集することができます。

編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「コンパス情報編集」を選択します（図 15-57）。

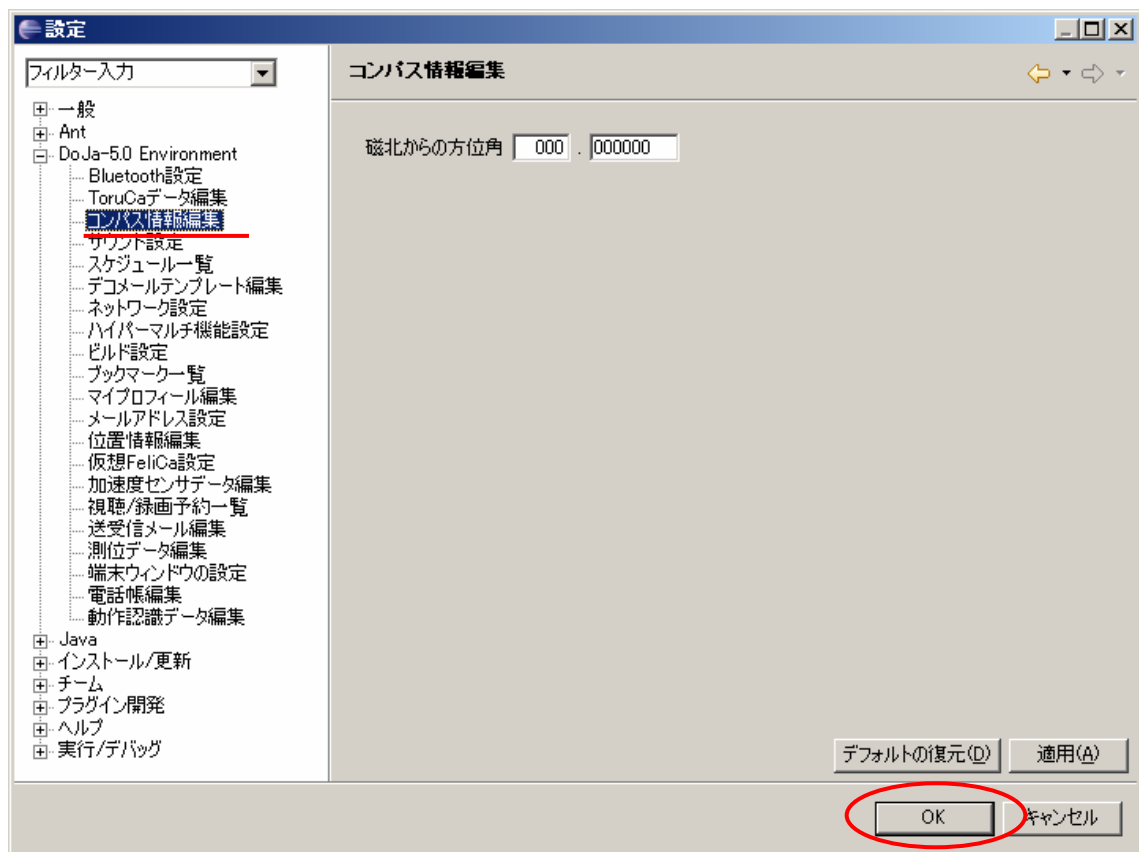


図 15-57 Eclipse 設定ウィンドウのコンパス情報編集画面

3. 各設定項目を設定します。
4. 「OK」を押すと、設定されます。

各設定項目の詳細については、「5.19. コンパス情報編集」を参照してください。

15.4.15. マイプロフィール情報編集

エミュレータのマイプロフィールの情報を編集することができます。
編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「マイプロフィール編集」を選択します（図 15-58）。

図 15-58 Eclipse 設定画面のマイプロフィール編集画面

3. 各設定項目を設定します。
4. 「OK」を押すと、設定されます。

各設定項目の詳細については、「5.20 マイプロフィール編集.」を参照してください。

15.4.16. ToruCa データ編集

エミュレータのトルカデータを編集することができます。

編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「ToruCaデータ編集」を選択します（図 15-59）。

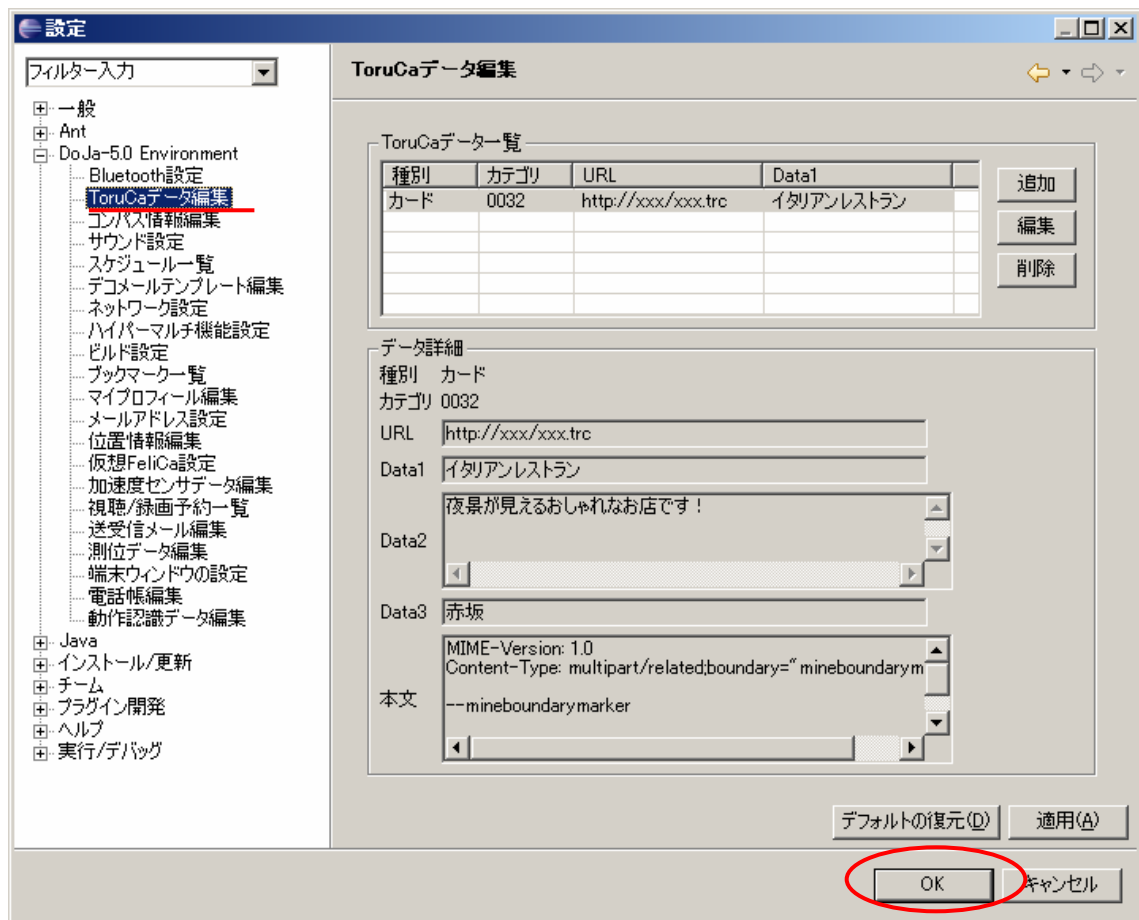


図 15-59 Eclipse 設定画面の ToruCa データ編集一覧画面

3. トルカを追加したい場合は「追加」、トルカを編集したい場合は一覧から編集したいトルカを選択して、「編集」ボタンを押します（図 15-60）。

ToruCa編集画面

データ詳細

種別: カード

ジャンル: ぐるめ

コード: 0032

バージョン: 2 . 0

IP-ID: 0000

カナID: アア

再配布識別子: 1

色ID: 1

有効期限: 061225

URL: http://xxx/xxx.trc

Data1: イタリアンレストラン ボーノボーノ

Data2: 夜景が見れるお洒落なお店です。

Data3: 赤坂

本文: <html><body><OBJECT declare id = "bono" data = "bono.jam" type = "application/><CENTER>夜景が見れるお洒落なイタリアンレストラン。</center></body></html>

設定 取消

図 1 5 - 6 0 ToruCa データ編集画面

4. 各設定項目を設定します。
5. 「設定」を押すと、ToruCa データ一覧に追加されます。
6. トルカを削除したい場合は、一覧からトルカを選択し、「削除」を押します。

各設定項目の詳細については、「5.2 1. トルカデータ編集」を参照してください。

15.4.17. デコメールテンプレート編集

エミュレータのデコメールテンプレートを編集することができます。

編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「デコメールテンプレート編集」を選択します（図 1 5-6 1）。

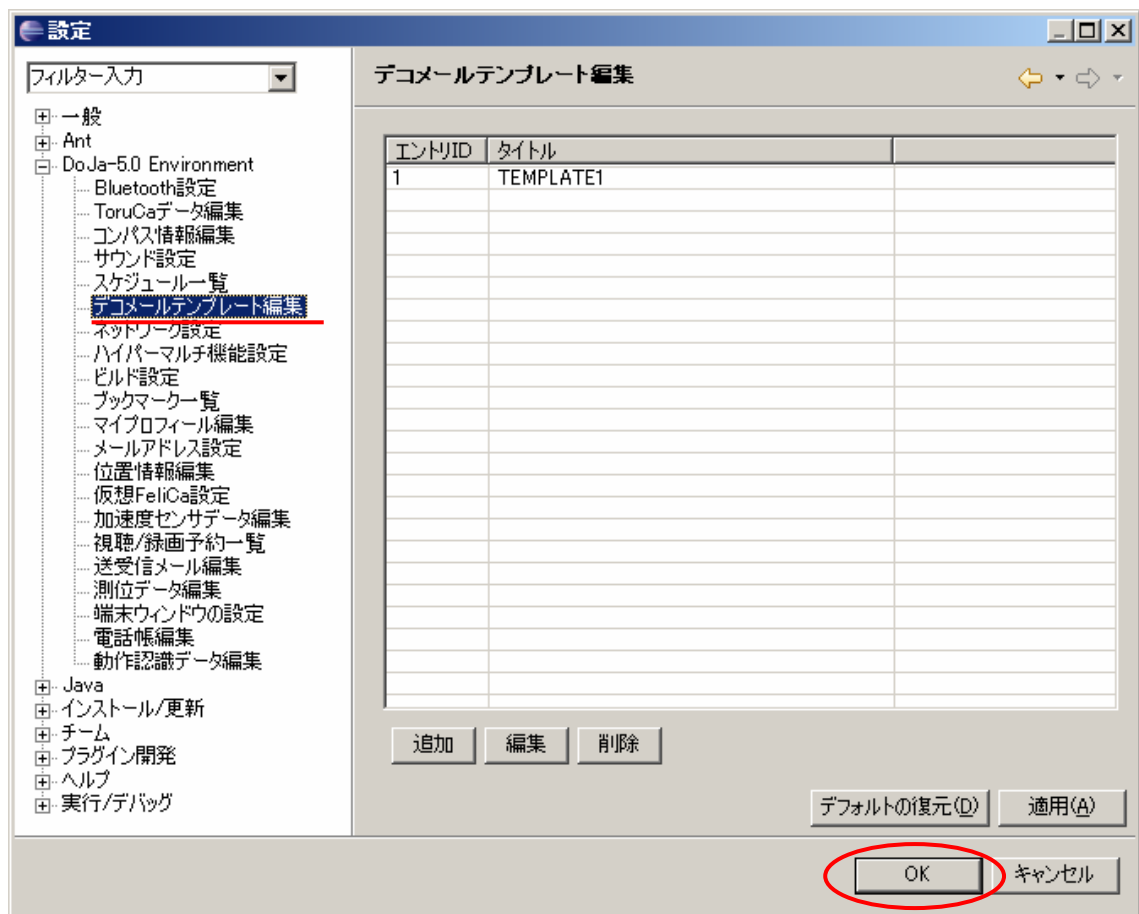


図 1 5-6 1 Eclipse 設定画面のデコメールテンプレート編集一覧画面

3. デコメールテンプレートを追加したい場合は「追加」、デコメールテンプレートを編集したい場合は一覧から編集したいデコメールテンプレートを選択して、「編集」ボタンを押します（図 1 5-6 2）。

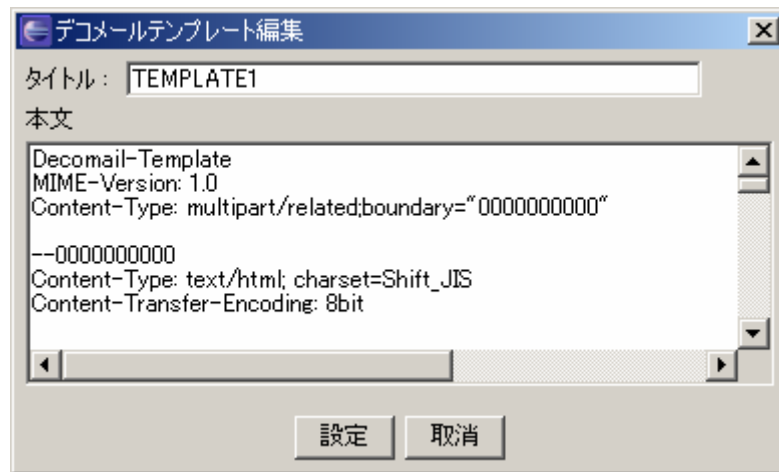


図 1 5 - 6 2 デコメールテンプレート編集画面

4. 各設定項目を設定します。
5. 「設定」を押すと、デコメールテンプレート一覧に追加されます。
6. デコメールテンプレートを削除したい場合は、一覧からデコメールテンプレートを選択し、「削除」を押します。

各設定項目の詳細については、「5.2 2. デコメールテンプレート編集」を参照してください。

15.4.18. Bluetooth 設定

TCP/IP上に接続された別のエミュレータと Bluetooth の API を利用した i アプリの動作を設定することができます。
設定をするには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「Bluetooth設定」を選択します（図 15-63）。

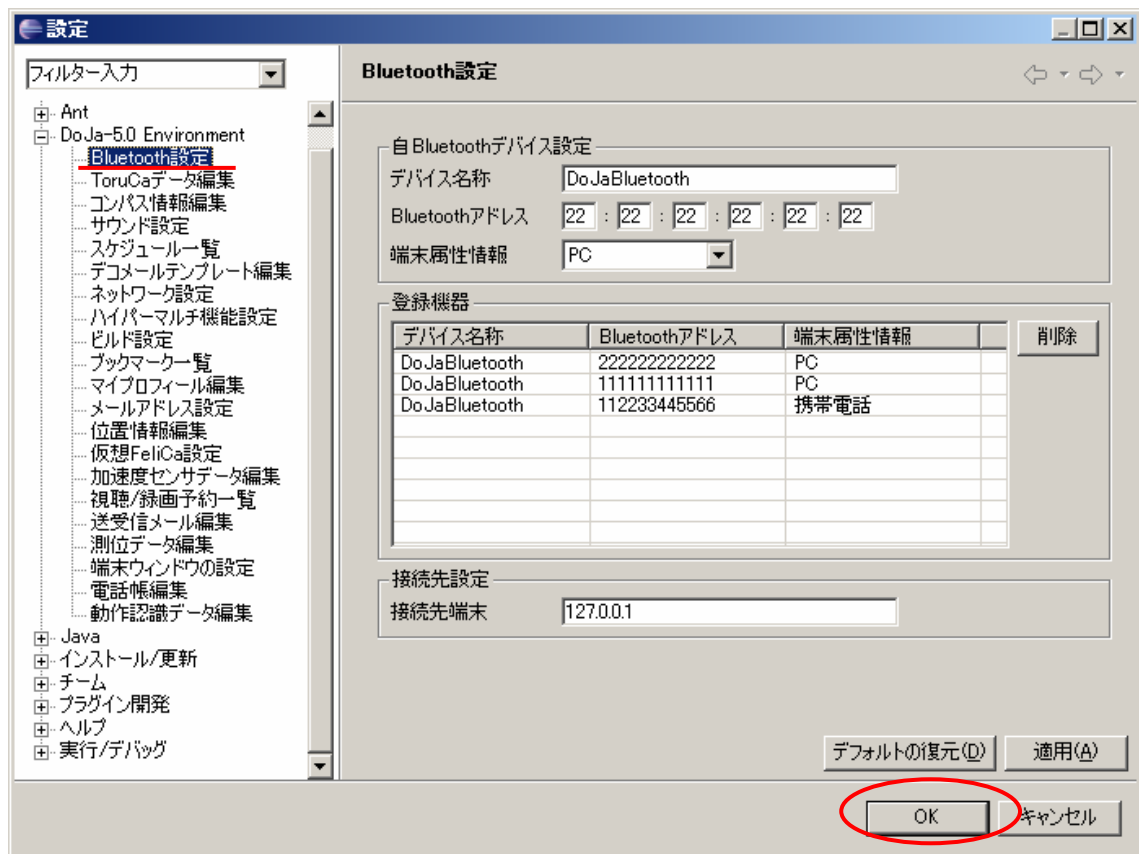


図 15-63. Bluetooth 設定画面

3. 各設定項目を設定します。
4. 登録機器は接続先によって自動的に登録されます。
不要な機器は、一覧から選択して「削除」を押してください。
5. 「OK」を押すと、設定されます。

各設定項目の詳細については、「5.9. Bluetooth設定」を参照してください。

15.4.19. 仮想 FeliCa 設定

TCP/IP上に接続された別のエミュレータと FeliCa の API を利用した i アプリの動作を設定することができます。
設定をするには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「仮想FeliCa設定」を選択します（図 15-64）。

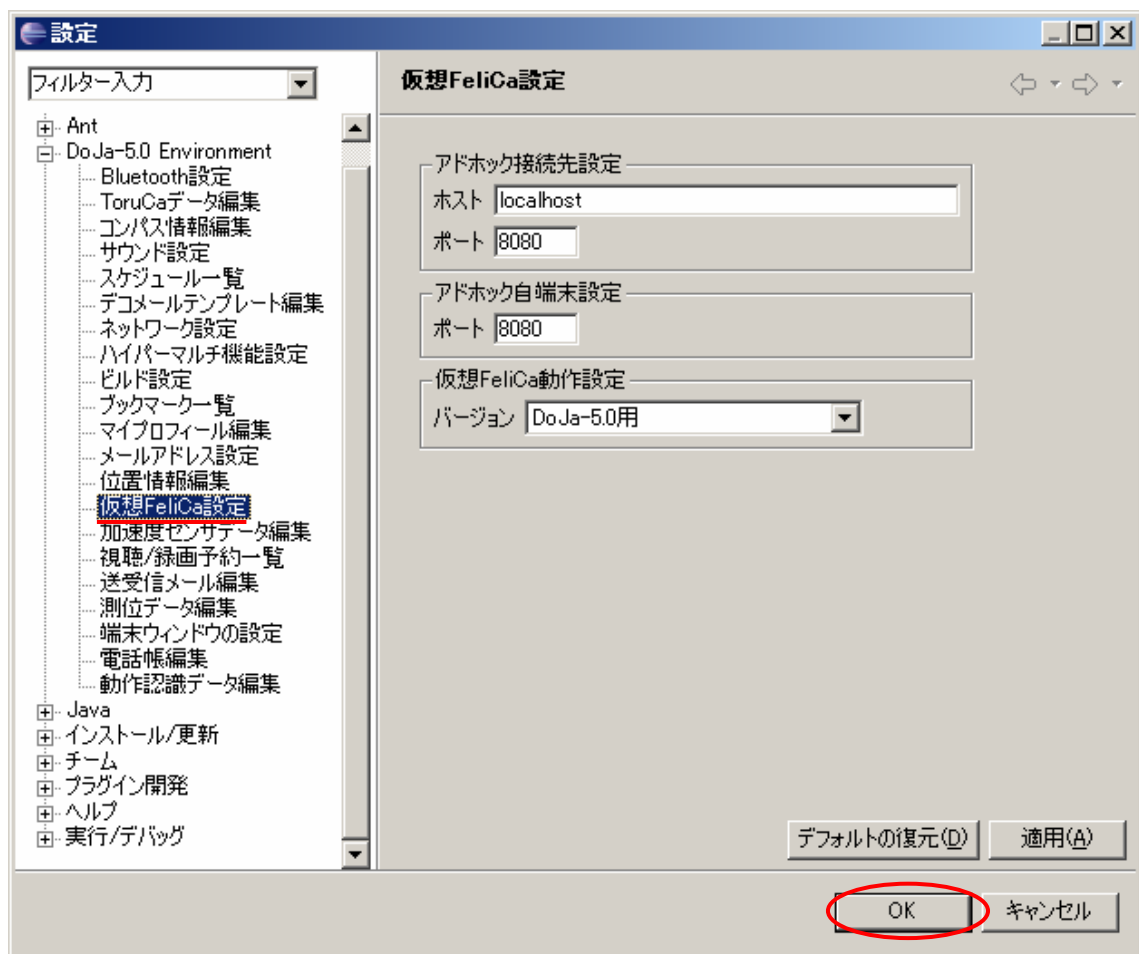


図 15-64. 仮想 FeliCa 設定画面

3. 各設定項目を設定します。
4. 「OK」を押すと、設定されます。

各設定項目の詳細については、「5.10. 仮想FeliCa設定」を参照してください。

15.4.20. 視聴/録画予約一覧

エミュレータの視聴/録画予約一覧を編集することができます。

表示するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「視聴/録画予約一覧」を選択します（図 15-65）。

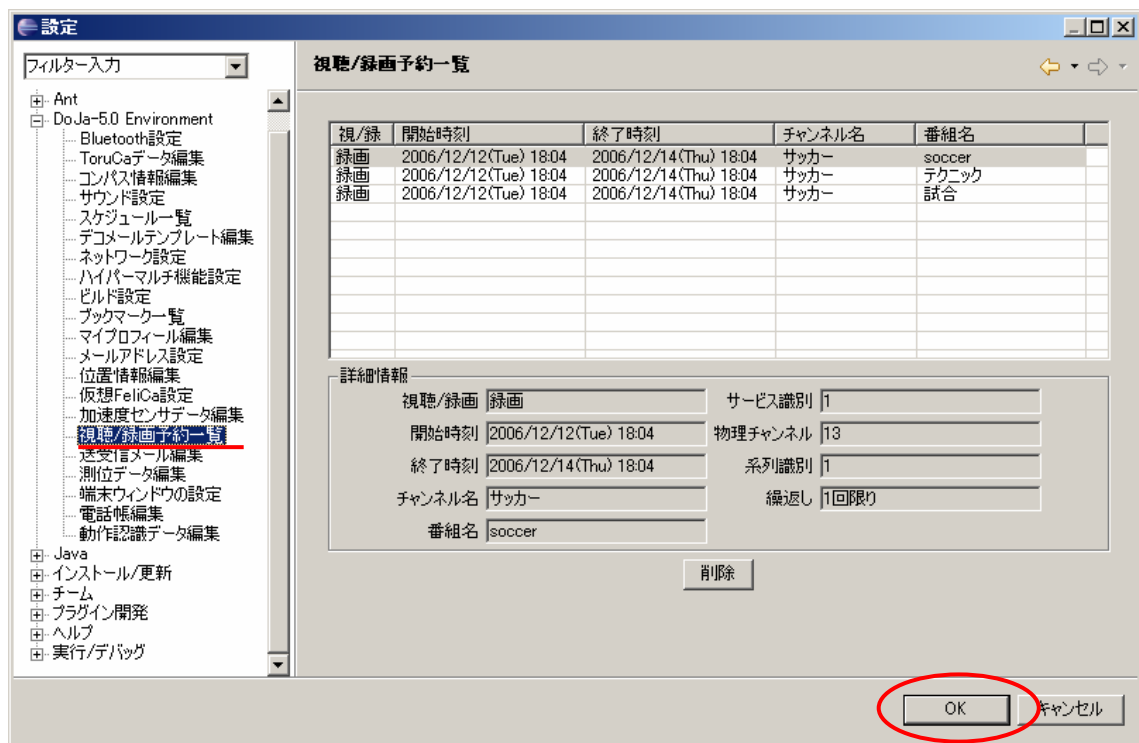


図 15-65. 視聴/録画予約一覧画面

3. 視聴/録画予約情報を削除するには、一覧から視聴/録画予約情報を選択して「削除」を押します。
視聴/録画予約情報ファイルが削除され、一覧から視聴/録画予約情報が削除されます。

15.4.21. ハイパーマルチ機能設定

ハイパーマルチ機能を設定することができます。

ハイパーマルチ機能を設定するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「ハイパーマルチ機能設定」を選択します（図 15-66）。

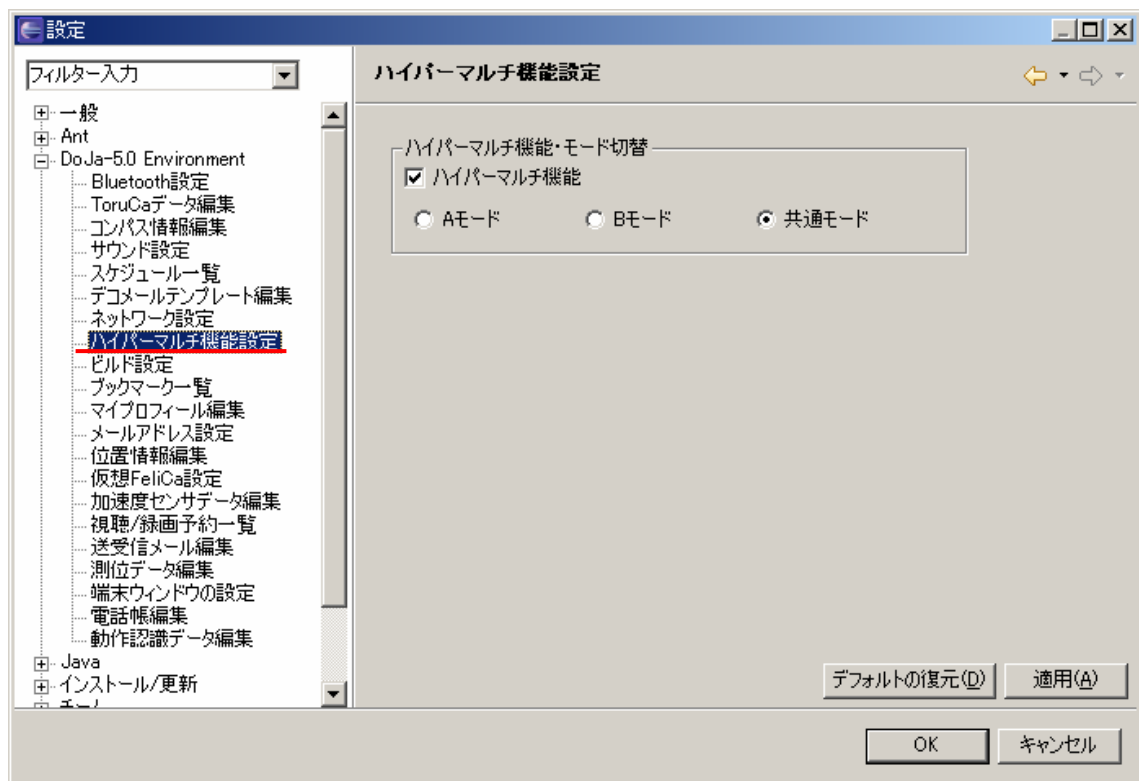


図 15-66 ハイパーマルチ機能設定画面

3. ハイパーマルチ機能にチェックを付けて、セパレートモードを選択します。

各設定項目の詳細については、「5.11. ハイパーマルチ機能設定」を参照してください。

15.4.22. 加速度センサデータ編集

加速度センサで取得するデータを編集することができます。
加速度センサデータを編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「加速度センサーデータ編集設定」を選択します（図 1 5 - 6 7）。

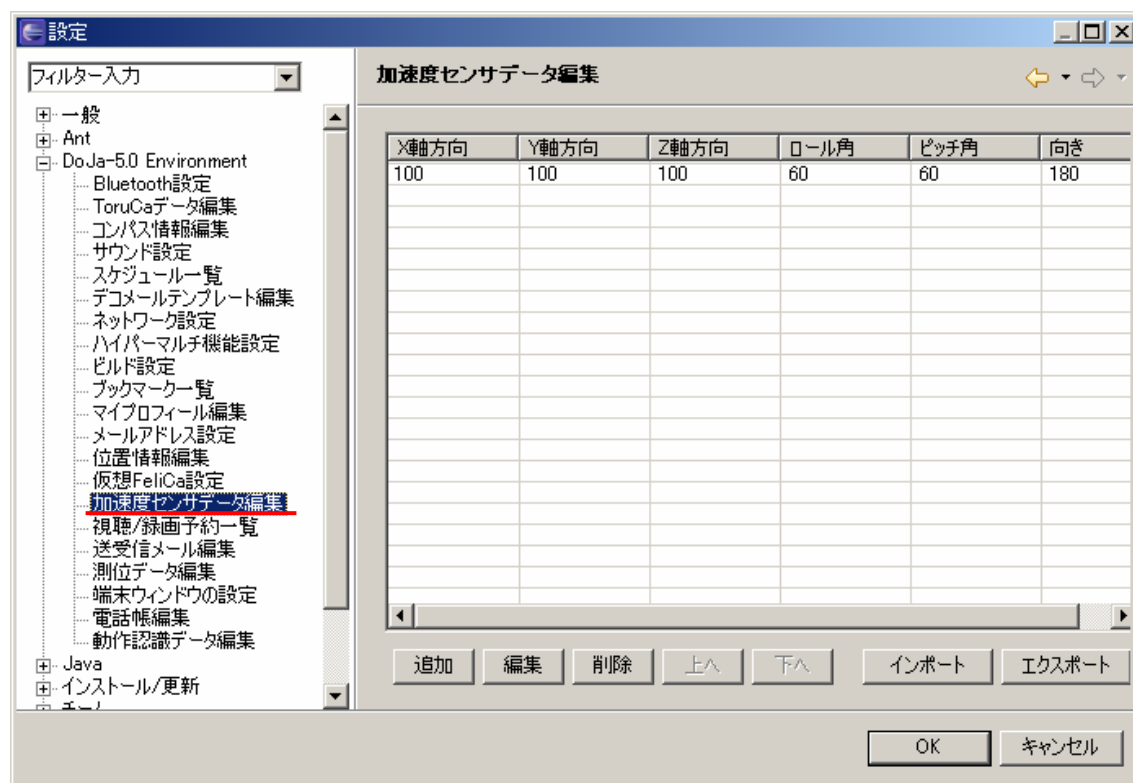


図 15-67 加速度センサデータ編集画面

3. 加速度センサデータを追加したい場合は「追加」、加速度センサデータを編集したい場合は一覧から編集したい加速度センサデータを選択して、「編集」ボタンを押します (図 15-68)。

加速度センサデータ設定

加速度

X軸方向 100 (mg)

Y軸方向 100 (mg)

Z軸方向 100 (mg)

端末の傾き

ロール角 60 (度)

ピッチ角 60 (度)

メインスクリーンの縦横の向き

向き 180 (度)

設定 取消

図 1 5 - 6 8 加速度センサデータ編集画面

4. 各設定項目を設定します。
5. 「設定」を押すと、加速度センサデータ一覧に追加されます。
6. 加速度センサデータの順序を変更したい場合は、順序を変更したい加速度センサデータを選択して「上へ」もしくは「下へ」を押します。
7. 加速度センサデータを削除したい場合は、一覧から加速度センサデータを選択し、「削除」を押します。
8. 加速度センサデータをCSVファイルからインポートしたい場合は、「インポート」を押します。
9. 加速度センサデータをCSVファイルにエクスポートしたい場合は、「エクスポート」を押します。

各設定項目の詳細については、「5.2 4. 加速度センサデータ編集」を参照してください

15.4.23. 動作認識データ編集

動作認識で取得するデータを編集することができます。
動作認識データを編集するには、次の手順に従ってください。

1. 「ウィンドウ」→「設定」を選択します。
2. Eclipse設定ウィンドウが表示されるので、「DoJa-5.0 Environment」→「動作認識データ編集設定」を選択します（図 15-69）。

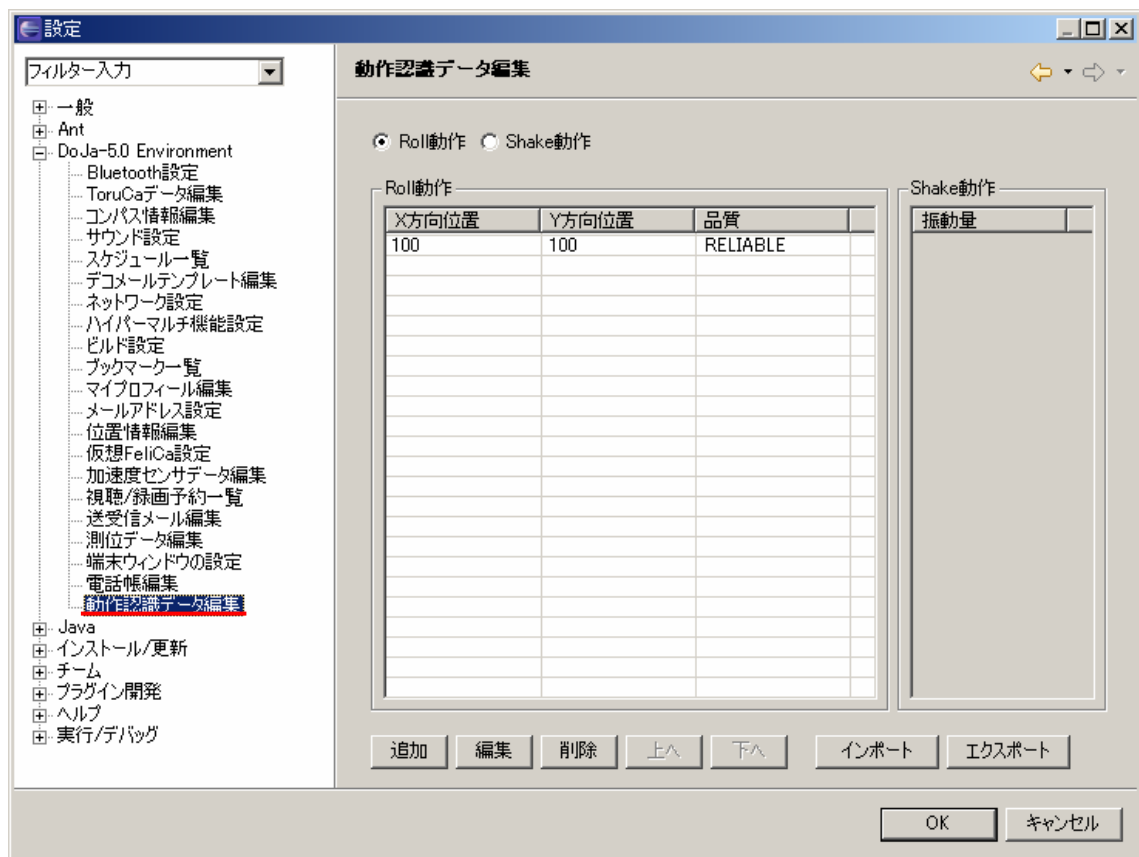


図 15-69 動作認識データ編集画面

3. 編集したい動作から Roll か Shake を選択します。
4. 動作認識データを追加したい場合は「追加」、動作認識データを編集したい場合は一覧から編集したい動作認識データを選択して、「編集」ボタンを押します（図 15-70）。



図 1 5-7 0 動作認識データ編集画面

5. 各設定項目を設定します。
6. 「設定」を押すと、動作認識データ一覧に追加されます。
7. 動作認識データの順序を変更したい場合は、順序を変更したい動作認識データを選択して「上へ」もしくは「下へ」を押します。
8. 動作認識データを削除したい場合は、一覧から動作認識データを選択し、「削除」を押します。
9. 動作認識データをCSVファイルからインポートしたい場合は、「インポート」を押します。
10. 動作認識データをCSVファイルにエクスポートしたい場合は、「エクスポート」を押します。

各設定項目の詳細については、「5.2 5. 動作認識データ編集」を参照してください。

15.5. 仮想 FeliCa データファイルビューワ

仮想 FeliCa データファイルビューワは、仮想 FeliCa データファイルを 16 進数でダンプします。

仮想 FeliCa データファイルビューワを使用すれば、アプリケーションから仮想 FeliCa データファイルにデータが適切に書き込まれているかどうかを確認する事ができます。

15.5.1. 仮想 FeliCa データファイルビューワの起動

1. メニューから「プロジェクト」→「DoJa-5.0」→「仮想 FeliCa データファイルビューワ…」を選択します。

「仮想FeliCaデータファイルビューワ」画面が表示されます（図 1 5 - 7 1）。

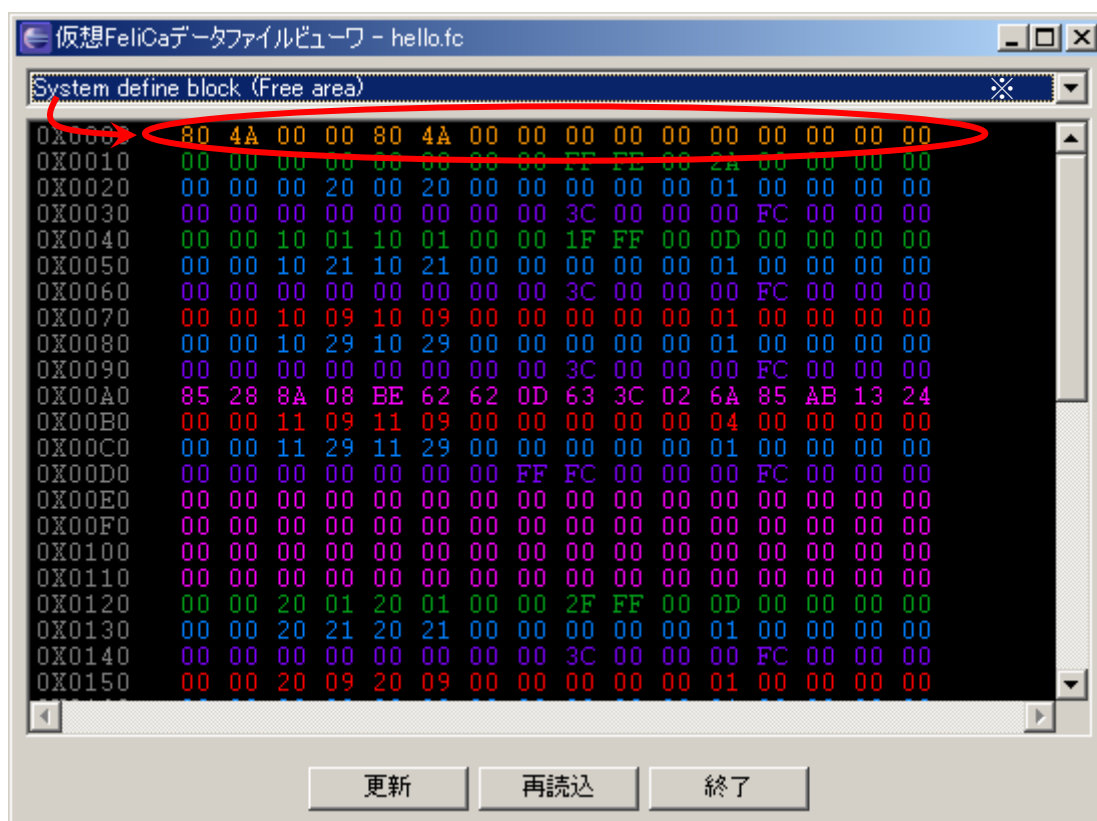
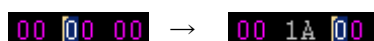


図 1 5 - 7 1 仮想 FeliCa データファイルビューワ画面

インストール直後のデータファイルは、データ領域を除き、適切な値が入っています。仮想FeliCaデータファイルの構造については、「1 0 . 3 2 . 仮想FeliCaデータファイル」を参照してください。

4. 現在読み込まれているプロジェクトの仮想 FeliCa データファイルの内容が 16 進数で表示されます。
5. 上段のプルダウン (※) から仮想 FeliCa データファイルの記憶領域を選択します。
該当する記憶領域のデータの色が変わります。
これにより、実際にアプリケーションから書き込まれた領域を参照する事ができます。
6. 値を変更したい場合は、変更したい所の値を選択し直接変更します。
仮想 FeliCa データファイルビューワでは、CPID Data Block, Data Block, PIN Data Block のみ値の変更が可能です。



※Area DefineBlock, Service Block 1, Service Block 2, PIN Service Block は変更しても無効です。

7. 「更新」を押すと、これまでの変更結果が仮想 FeliCa データファイルに書き出されます。
8. 「再読込」を押すと、仮想 FeliCa データファイルの内容を再読込みし、仮想 FeliCa データファイルビューワの表示時点もしくは「更新」を押した時点の状態に戻ります。
9. 「閉じる」を押すと仮想 FeliCa データファイルビューワを終了します。

仮想 FeliCa データファイルビューワで表示されたフリー領域を扱うアプリケーションを利用するには、次の条件が満たされている必要があります：

- CPID Data Blockが、A D FのFeliCaCPIDと同じ値でなければなりません（「4.7. アプリケーションディスクリプタファイル (ADF) の編集」参照）。
- PIN Data Blockの先頭 4 バイトが、A D FのFeliCaPINの値を 16 進数表現した値でなければなりません（「4.7. アプリケーションディスクリプタファイル (ADF) の編集」参照）。

これは、A D Fの設定で FeliCaCPID、FeliCaPIN を設定した場合に自動的に関連付けをおこなう事ができます（「1 5.3.6.2. FeliCaフリー領域の関連付け」参考）。

設定手順は：

1. 仮想 FeliCa データファイルビューワの上段にあるプルダウンから CPID Data Block を選択します。
2. CPID を格納する場所が桃色で表示されるので、この部分にカーソルを当て、ADF に設定した FeliCaCPID と同じ値に変更します。
3. 仮想 FeliCa データファイルビューワの上段にあるプルダウンから PIN Data Block を選択します。
4. PIN を格納する場所が紫色で表示されるので、この部分にカーソルを当て、先頭 4 バイトに ADF に設定した FeliCaPIN の値を 16 進数表現で設定します。例えば、ADF に設定した FeliCaPIN の値が 12345678 の場合は、PIN Data Block の先頭 4 バイトを 00BC614E に変更します。
5. 仮想 FeliCa データファイルビューワの「更新」ボタンを押します。
6. 「閉じる」ボタンを押します。

【注意事項】

- 仮想 FeliCa データファイルビューワを表示している状態で、アプリケーションから仮想 FeliCa データファイルヘデータを書き込んだ場合は、「再読込」ボタンを押してください。
- アプリケーションが仮想 FeliCa データファイルにアクセスしている状態で、仮想 FeliCa データファイルを編集した場合、アプリケーションで例外が発生する可能性があります。
- フリー領域を利用するアプリケーションで、ADF の FeliCaCPID、FeliCaPIN の値を設定して、仮想 FeliCa データファイルを作成しても、フリー領域の CPID Data Block、PIN Data Block には自動的に反映されません。

15.5.2. 仮想 FeliCa データファイルビューワの表示色

仮想 FeliCa データファイルビューワの表示で、ウィンドウ内の色は下記のようになっています。

橙 : SystemDefineBlock
 緑 : AreaDefineBlock
 赤 : Service Block
 水色 : PIN Service Block
 桃 : CPID Data Block および Data Block
 紫 : PIN Data Block
 灰色 : 未定義領域

15.5.3. ブロックの構成

仮想 FeliCa データファイルビューワでは、各ブロックは下記のような構成になっています。

1 ブロックは 16 バイトで、以下のようなバイト列になっています。

D0	D1	D2	D3	．．．．．	D13	D14	D15
----	----	----	----	-------	-----	-----	-----

D = バイト

DoJa-4.1(以前)用の場合

- SystemDefineBlock
D0～D1 : System Code
- AreaDefineBlock
D0～D1 : Area Code
D2～D3 : End Service Code
- Service Block
D0～D1 : Service Code

DoJa-5.0 用 の場合

- SystemDefineBlock
D0～D1 : System Code
- AreaDefineBlock
D0～D3 : Area Code
D6～D9 : End Service Code
- Service Block
D0～D3 : Service Code

- 扱う FeliCa データが DoJa-4.1 用 か DoJa-5.0 用 かは、仮想 FeliCa 設定のバージョンで変更できます（「15.4.19. 仮想 FeliCa 設定」参照）。

仮想 FeliCa データファイルビューワで編集を行う場合のエンディアン表記については、全てビッグエンディアンになっています。

15.6. スクラッチパッド

15.6.1. スクラッチパッド管理ツール

スクラッチパッド管理ツールは、アプリケーションのスクラッチパッドファイルを 16 進数でダンプして編集するためのツールです。

スクラッチパッド管理ツールを使用すれば、スクラッチパッドのデータを作成したり、アプリケーションからスクラッチパッドにデータが適切に書込まれているかどうかを確認したりすることができます。

Eclipse のスクラッチパッド管理ツールは、複数のスクラッチパッドを 1 つの領域で表示します。各スクラッチパッドで編集したい場合は、*ia ppliTool* をご利用ください。

15.6.2. スクラッチパッド管理ツールの起動

1. メニューから「プロジェクト」→「DoJa-5.0」→「スクラッチパッド...」を選択します。
2. 「スクラッチパッド管理ツール」画面が表示されます（図 15-72）。

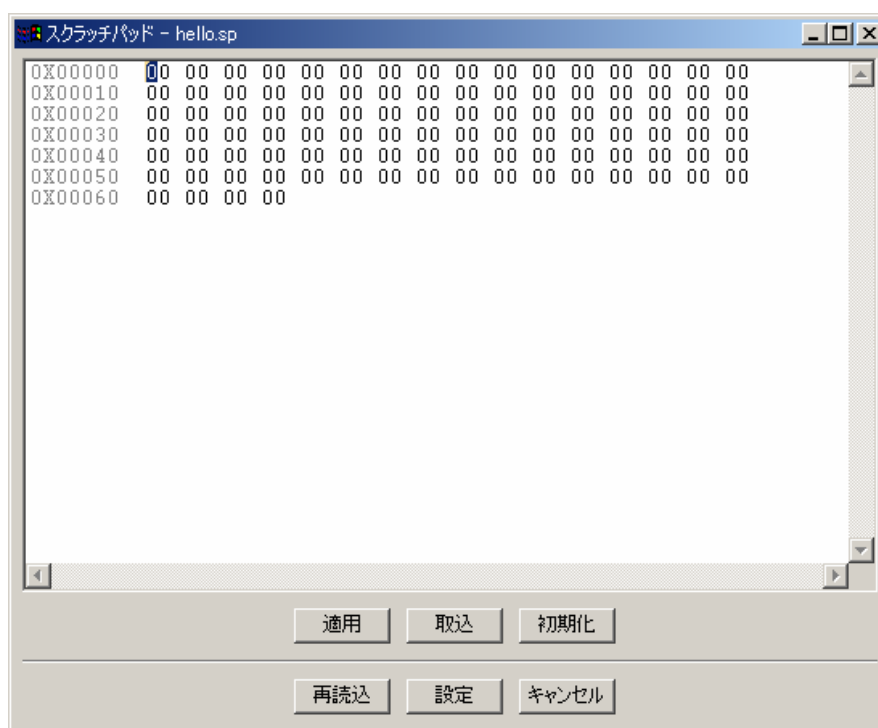


図 15-72 スクラッチパッド管理ツール画面

3. 現在、読み込まれているプロジェクトのスクラッチパッドの内容が、16 進数で表示されます。

4. 「適用」を押すと、これまでの変更結果をスクラッチパッドファイルに書き込みます。
5. 「取込」は、カーソルの位置から外部のファイルを取り込むことができます。
まず、取り込みたい位置を選択します（図 1 5-7 3）。

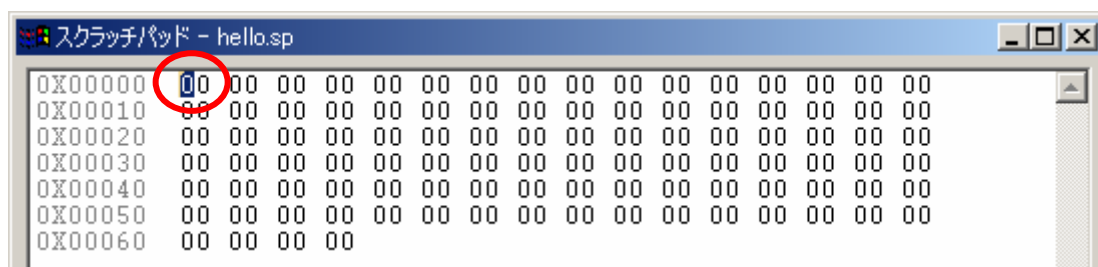


図 1 5-7 3 スクラッチパッド 置換開始位置選択

「取込」を押すと、「スクラッチパッドファイル選択」ダイアログが表示されるので、取り込みたいファイルを選択し、「開く」を押します（図 1 5-7 4）。

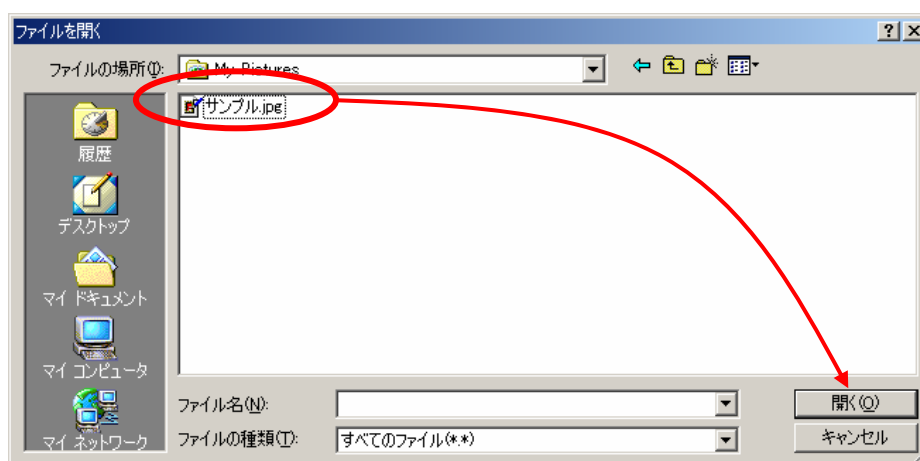


図 1 5-7 4 置換ファイル選択ダイアログ

カーソルの位置から選択したファイルの内容が取り込まれます（図 1 5-7 5）。
ファイルのサイズがスクラッチパッドのサイズを超えた場合は途中で切り捨てられ警告されます。

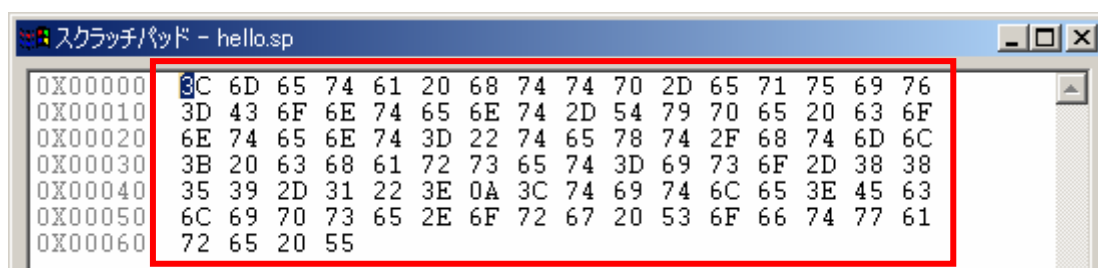


図 1 5-7 5 置換後のスクラッチパッド

6. 「初期化」を押すと、スクラッチパッドの表示内容を0で初期化します。
7. 「再読込」を押すと、スクラッチパッドファイルの内容を再読込みし、スクラッチパッド管理ツールの表示時点もしくは、「適用」を押した時点の状態に戻ります。
8. 「設定」を押すと、これまでの変更結果をスクラッチパッドファイルに書き込んで終了します。
9. 「キャンセル」を押すと、スクラッチパッドファイルは変更せずに終了します。

15.6.3. スクラッチパッドを全て初期化する

メニューから「プロジェクト」→「DoJa-5.0」→「スクラッチパッドを全て初期化」を選択することで、現在読込まれているプロジェクトのスクラッチパッドを全て削除することができます。スクラッチパッドが存在しない場合、「スクラッチパッドがありません」というダイアログが表示されます。

15.6.4. スクラッチパッドでJarファイルを扱う場合の注意事項

スクラッチパッドにJarを保存してアプリケーションで扱う場合、JarファイルのサイズをURLのlengthで指定するか、スクラッチパッドのサイズ（ADFのSPsize（「4.7. アプリケーションディスクリプタファイル（ADF）の編集」参照。））とJarファイルのサイズが同じでないとフォーマットエラーになります。

付録.1 エミュレーション端末の設定

エミュレーション端末設定ファイルを書き換えることでエミュレーション端末の動作を変更する事が可能です。

エミュレーション端末設定ファイルは、"`lib\skin\端末名\skin.conf`"です。

ただし、下記の情報以外を書き換えることは、i アプリの動作に支障を及ぼす可能性があるもので、お勧めできません。

1. サウンドライブラリの設定

サウンドライブラリのプロパティを指定できます。

`sound_param_audio3d_resources` = [3 D 仮想音源数]

指定値：0 ～ 8

未指定：4

`sound_param_adpcm_resources` = [ADPCM 同時発音数]

指定値：0 ～ 4

未指定：4

`sound_param_block_num` = [ブロック数(ストリーミングバッファサイズ)]

指定値：制限なし

未指定：2

`sound_param_wait_time` = [DryTrack の 3 DTrack に対する遅延調整時間]

指定値：0 ～ 500

未指定：0

※ サウンドライブラリの設定値を指定する場合は、新たにファイルに追加してください。

用語の説明

用語	説明	関連ページ
iαppliTool	エミュレータの機能設定画面。	9,26
エミュレーション端末	エミュレータの携帯端末画面。	26,204,203,196
i アプリ	i モード携帯電話で動作する J a v a アプリケーション(ソフトウェア)。	30
トラステッド i アプリ	i モード携帯電話の情報 (メールや発着信履歴など) を扱う i アプリ。 i モードメニューに掲載されていないコンテンツ提供者は扱えません。使いたい機能がトラステッドであるかどうかは、別途、A P I リファレンスを確認してください。 →「トラステッド i アプリのみこのメソッドを呼び出すことができます。」というような記述がある場合には、トラステッドになります。	30,125
i アプリ D X	項番 2 に同じ。	-
D o J a	i アプリの J a v a 仕様 (A P I) 。 i アプリでは、C L D C 1. 1 と D o J a の A P I が扱えます。	9
C L D C	Connected Limited Device Configuration D o J a が動作する組み込み機器用の J a v a 環境 (J2ME)。	9
J 2 S E J D K	Java(TM) 2 SDK, Standard Edition Sun Microsystems 社 の Java2 の開発環境で、Sun Microsystems 社から取得してインストールできます。 J2SE を使えるようにするには、Windows の環境変数 JAVA_HOME に J2SE のインストール先を設定しておく必要があります。 本エミュレータは、1.4.2 を保証しております。	9
D i r e c t X	Microsoft 社の Windows 用拡張 A P I で、Microsoft 社から取得してインストールできます。 → Windows Update	9
待ち受け i アプリ	待ち受け画面で動作する i アプリ。 折りたたみ状態では省電力で動くため、待ち受け画面上で常に起動しておくことができます。	66
J A M	Java Application Manager i アプリのダウンロードや起動を制御する機能。	52
A D F	Application Descriptor File i アプリの起動や動作の情報が格納されたファイルで、JAM が最初に読み込みます。 i アプリは、ADF がないと動かせません。	52
サスペンド	i アプリが通話中などで中断している状態。	66,204
レジューム	サスペンドから復帰した状態。 レジュームは、i アプリにイベントとして渡されます。	66,204
休眠状態	待ち受け i アプリで起動しているが動いていない省電力の状態。	66,204
活性化状態	待ち受け i アプリで通常起動の状態 (i アプリの設定画面など) 。 待受ボタンで切り替え。	66,204

非活性化状態	待ち受け i アプリで待ち受けの状態。 待受ボタンで切り替え。	66,204
スクラッチパッド	i アプリのデータを永続的に保存しておく領域。	30,52,214,263
ヒープ	i アプリが扱うメモリ（一時的な保存領域）。 ヒープには、DoJa が使うメモリと携帯ネイティブ (電話帳やメール)が使うメモリがあります。 ヒープのサイズは機種によって異なります。	135
コード認識	QR コードやバーコードの認識機能。	89,250
位置情報	GPS で測位した位置履歴情報。 ※GPS 機能が搭載された携帯電話機のみ利用可。	94,97,100,159, 144,254
測位データ	GPS で測位した結果。 ※GPS 機能が搭載された携帯電話機のみ利用可。	162,255
FeliCa	ソニー社の非接触 IC カード機能。 (おサイフケータイ、モバイル FeliCa)	52,63,64,141, 219,262
FeliCa アドホック	FeliCa を用いた無線通信機能。 ※ DoJa-5.0 から導入	103,141,258
トルカ	FeliCa を用いた電子カード機能。 ※ DoJa-4.1 から導入	107,169,252
デコメール	i モードメールをデコレーションした HTML 形式の メール。	172,256
リモート制御	i アプリをリモートから制御（主にサービスの停止） する機能。	130131
UART	Universal Asynchronous Receiver Transmitter PC のシリアル回線などでデータ通信をする機能。	136
O B E X	OBject EXchange 無線通信のプロトコル。	122,268
I r D A	Infrared Data Association 赤外線通信の規格。	52,268
Bluetooth	携帯端末向けの無線通信機能。 ※Bluetooth 機能が搭載された携帯電話機のみ利用可。	138,258
D T V	Digital Television デジタル変調・圧縮の技術を利用したテレビ放送。	116,259
BML	Broadcast Markup Language データ放送向けのページ記述言語。	112,259
BML ブラウザ	BML で記述されたページを閲覧するためのアプリケ ーション。	112,259
動作認識	カメラデバイスで撮影された画像の移動量を認識する 機能。揺れや傾きを計測して補正するのに用いる。	179
加速度センサ	携帯端末の移動速度の変化を計測する回路素子。	176
Sun Java Studio Enterprise	Sun Microsystems 社 の J a v a 統合開発環境。 (フリー)	10,269
N e t B e a n s	Sun Microsystems 社 の推奨するオープンソースの J a v a 統合開発環境。(フリー)	10,269
E c l i p s e	幅広く使われているオープンソースの統合開発環境。 (フリー)	10,284