

Star-2.0 API i α ppli Development Kit ユーザーズガイド

平成 24 年 8 月 31 日
株式会社 NTT ドコモ

改版履歴

版数	項目	種別	内容	
1.00	-	新規	-	2011 年 10 月 Star-2.0 向け 制定 <iappli Development Kit for Star-2.0>
1.01	-	変更	改版内容	開発支援情報画面の各グラフの色が全て黒色で表示される場合がある問題に対応
			改版理由	不具合対処
	-	変更	改版内容	描画最適版をインストール可能な環境であるにもかかわらずインストーラで「インストールプログラムの選択」画面が表示されない場合がある問題に対応
			改版理由	不具合対処
1.01 (2012/8/31 版)	9.8	追加	改版内容	ドコモが発行したユーザ証明書を使用する方法で署名しても i アプリで利用できない旨を追記。 ※この変更はユーザズガイドのみで、i appliTool 本体の変更はありません。
			改版理由	FirstPass サービス終了に伴う修正

目次

第1章 はじめに	13
1.1. iappli Development Kit for Star-2.0 の動作環境.....	13
1.1.1. iappliTool について	13
1.2. iappli Development Kit for Star-2.0 の概要.....	14
1.2.1. iappli Development Kit for Star-2.0 の利点.....	14
1.2.2. 統合開発環境の利用	14
1.2.3. エミュレーションの範囲	15
1.3. iappli Development Kit for Star-2.0 による i アプリの開発.....	18
第2章 インストール	20
2.1. インストールする前に	20
2.2. インストール手順	21
2.3. アンインストール手順	29
第3章 iappliToolの起動	30
3.1. ビルドエラービュー	31
3.2. プロジェクト・エクスプローラ	32
第4章 i アプリの作成	34
4.1. フルアプリとミニアプリ	35
4.1.1. ミニアプリの画面表示構成	36
4.1.2. フル+ミニ、ミニ+フルアプリ	37
4.2. i アプリの作成から起動まで	38
4.3. プロジェクトの新規作成	39
4.3.1. ソースファイルの作成	41
4.3.2. ソースコードの修正	45
4.3.3. ソースエディタの設定	46
4.4. プロジェクトのビルドと通常起動	48
4.4.1. ダウンロード用HTMLファイル	52
4.4.2. i モード携帯電話で i アプリ実行	53
4.5. プロジェクトの読み込み	54
4.5.1. 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	56
4.6. プロジェクトの設定	57
4.7. ファイル配置カスタマイズ	62
4.8. アプリケーションディスクリプタファイル (ADF) の編集.....	63
AppType.....	64
AppName.....	64
AppVer	64
PackageURL.....	64

InnerPackageName.....	65
InnerAppSize.....	65
AppSize	66
ConfigurationVer.....	66
ProfileVer.....	66
SPsize	67
AppClass	68
AppParam.....	68
LastModified	69
UseNetwork.....	69
TargetDevice	70
LaunchAt.....	70
UseTelephone.....	71
UseBrowser	71
UseDTV.....	71
LaunchByMail.....	71
LaunchByBrowser.....	72
LaunchByToruCa	72
LaunchByBML.....	72
AllowPushBy.....	73
AppTrace	73
DrawArea	74
Pallet.....	75
GetSysInfo.....	76
TrustedAPID	76
TrustedLmd	76
PackageHashCode	76
LaunchByApp	77
MessageCode.....	77
AccessUserInfo.....	77
GetUtn	78
LaunchApp	78
FeliCaCPID	78
FeliCaPIN.....	79
AppIcon.....	79
UseStorage.....	79
CPBindingAuth	80
CPName.....	80
GetLocationParam.....	80
AppMainTitle	81
DeniedMultiApp.....	81
UseMSO	82
MenuTitleImage	82
AppLabel	82
LaunchBySchedule.....	83
LaunchByDataBox	83
RestrictedAudioOutput.....	83
VmOption.....	83
LaunchByFaceApp.....	84
StartingAppImage.....	84
4.8.1. FeliCaCPIDの設定	85
4.8.2. FeliCaフリー領域の関連付け.....	86
4.8.3. LastModified項目の表記方法.....	87
4.8.4. アイコン・タイトル画像、起動時画像の確認方法	88
4.9. トラストッド動作の設定	91

4.10. 受信フォルダ起動	95
4.10.1. 受信フォルダ起動手順	95
4.11. 送信フォルダ起動	95
4.11.1. 送信フォルダ起動手順	95
4.12. 未送信フォルダ起動	96
4.12.1. 未送信フォルダ起動手順	96
4.13. 初回起動	96
4.13.1. 初回起動手順	96
4.14. タイマ起動	97
4.14.1. タイマ起動手順	97
4.15. 外部機器起動	98
4.15.1. 外部機器起動手順	98
4.16. メーラ起動	99
4.16.1. メーラ起動手順	99
4.17. ブラウザ起動	101
4.17.1. ブラウザ起動手順	101
4.18. コードリーダー起動	104
4.18.1. コードリーダー起動手順	104
4.18.2. CodeReaderの実行手順	106
4.19. 位置情報起動	107
4.19.1. 位置情報起動手順	107
4.20. 位置情報画像起動	109
4.20.1. 位置情報画像起動手順	109
4.21. 位置情報電話帳起動	111
4.21.1. 位置情報電話帳起動手順	111
4.22. アドホック起動	113
4.22.1. アドホック起動手順	114
4.22.2. i アプリが起動中の場合	115
4.23. トルカ起動	116
4.23.1. トルカ起動手順	116
4.24. 削除時起動	119
4.24.1. 削除時起動手順	119
4.25. BML起動	120
4.25.1. BML起動手順	120
4.26. DTV起動	122
4.26.1. DTV起動手順	122
4.27. 地図プラットフォーム起動/ナビ起動	125
4.27.1. 地図プラットフォーム起動/ナビ起動手順	125
4.28. 招集メッセージ起動	126
4.28.1. 招集メッセージ起動手順	126

4.29. コピー文字列起動	127
4.29.1. コピー文字列起動手順	127
4.30. スケジューラ連携起動	128
4.30.1. スケジューラ連携起動手順	128
4.31. データBOX画像連携起動	129
4.31.1. データBOX画像連携起動手順	129
4.32. Bluetooth連携起動	131
4.32.1. Bluetooth連携起動手順	131
4.32.2. Bluetooth HDP機器からの接続要求による連携起動	132
4.33. パッケージ起動	133
4.33.1. パッケージ起動手順	133
4.34. MyFACE起動	134
4.34.1. MyFACE起動手順	134
4.35. Widget View 表示	137
4.36. Widget View起動	139
4.37. 各種起動方法による画面遷移	140
第 5 章 i アプリBookmarkの作成	142
5.1. i アプリBookmarkの作成手順	143
5.2. プロジェクトの作成	144
5.2.1. プロジェクト新規作成	144
5.2.2. 既存の i アプリプロジェクトから作成	146
5.3. アプリケーションディスクリプタファイル (ADF) の編集	148
AppName	149
PackageURL	149
LaunchByMail	149
LaunchByBrowser	149
LaunchByToruCa	149
LaunchByBML	149
AllowPushBy	149
AppIcon	149
AppMainTitle	149
BookmarkURL	150
BookmarkAppURL	150
BookmarkPreinType	150
BookmarkSize	150
LaunchBySchedule	150
LaunchByFaceApp	150
5.4. プロジェクトのビルド	152
第 6 章 アプリケーション動作環境設定	153
6.1. ネットワークの設定	153
6.1.1. プロキシサーバの設定	155
6.2. 全般設定	156

6.3. メールアドレスの設定	159
6.4. サウンド設定	161
6.5. UART用COMポートの設定	164
6.6. Bluetooth設定	166
6.6.1. Bluetooth設定	166
6.6.2. Bluetooth HDP設定	169
6.7. 仮想FeliCa設定	173
6.8. ハイパーマルチ機能設定	175
第7章 ネイティブデータ編集	177
7.1. 電話帳の編集	177
7.2. グループの編集	181
7.3. ブックマーク一覧	183
7.4. 送受信メールの編集	184
7.4.1. SMSの扱いについて	188
7.4.2. メールを送受信エミュレート機能	190
7.5. 位置情報編集	191
7.6. 測位データ編集	194
7.6.1. 測位データのインポート・エクスポート	197
7.7. コンパス情報編集	199
7.8. トルカデータ編集	201
7.9. 視聴/録画予約一覧	204
7.10. 加速度センサデータ編集	206
7.11. 動作認識データ編集	209
7.12. 音声認識データ編集	212
7.13. 歩数計データ編集	216
7.14. 身体活動量計データ編集	220
7.15. オートGPS制御情報一覧	223
7.16. ジャイロセンサデータ編集	225
7.17. 身体情報編集	228
7.18. 脈拍計データ編集	230
7.19. 音楽コンテンツ編集	232
7.20. プレイリスト編集	235
第8章 エミュレータ環境設定	237
8.1. ビルドの設定	237

8.1.1. スクリプトの実行	237
8.1.2. ターゲットプロファイルの設定	240
8.2. ADF/トラステッド動作設定チェック処理の設定	241
8.3. ショートカットキーの設定	243
8.4. ヘルプの設定	246
8.5. エミュレータ性能設定	248
8.6. エミュレーション端末ウィンドウの設定	251
8.6.1. オプション機能対応設定	254
8.7. タイトル設定	259
8.8. スクリーンショット保存の設定	260
8.9. 開発支援情報表示設定	261
8.10. FeliCaPush通知パラメータ設定	263
8.11. 起動時にプロジェクトを読み込む	263
8.12. .jam に関連付ける	263
8.13. コンソールクリア	263
8.14. エミュレーション端末の選択	264
8.15. エミュレーション端末のスケール変更	266
8.16. エミュレーション端末の操作	267
8.16.1. 端末状態ウィンドウの操作	278
8.16.2. オプションキーウィンドウの操作	280
8.16.3. 開発支援情報	281
8.16.4. 3Dサウンド表示	282
8.16.5. 音楽再生状況表示	283
第 9 章 ツール	285
9.1. スクラッチパッド管理ツール	285
9.1.1. スクラッチパッドファイル	285
9.1.2. スクラッチパッド管理ツールの起動	285
9.1.3. スクラッチパッドでJARファイルを扱う場合の注意事項	288
9.2. スクラッチパッドを全て初期化する	289
9.3. 仮想FeliCaデータファイルビューワ	290
9.3.1. 仮想FeliCaデータファイル	290
9.3.2. 仮想FeliCaデータファイルビューワの起動	290
9.3.3. 階層表示	291
9.3.4. バイナリ表示	293
9.3.5. フリー領域を扱う i アプリの利用	294
9.4. Javadoc 作成ツール	296
9.5. ADFチェックツール	299
9.6. スクリーンショット	301

9.7. i アプリライブラリ	302
9.7.1. i アプリライブラリの作成	302
9.7.2. i アプリライブラリの利用	309
9.8. CPBindingAuth署名付与ツール.....	312
9.8.1. ドコモが発行したユーザ証明書を使って署名する方法	312
9.8.2. ユーザが発行した証明書を使って署名する方法	313
第 10 章 iappliToolの終了	315
第 11 章 デバッグトレース	316
第 12 章 i アプリ連携	318
第 13 章 フルミニ連携	319
第 14 章 ステレオ表示機能	320
14.1. 表示方式	320
14.2. サイドバイサイド表示	321
14.2.1. サイドバイサイド表示方式切り替え	321
14.2.2. 解像度比設定による表示切り替え	322
14.3. 3D表示	324
14.4. ステレオ表示のスタイル切替	325
第 15 章 ファイルの記述方法	326
15.1. メーラ起動ファイル	328
15.2. バウンダリデータファイル	328
15.3. メールアドレスファイル	328
15.4. ブラウザ起動HTMLファイル	328
15.5. 着信履歴ファイル	329
15.6. メールファイル	329
15.7. 電話帳ファイル	330
15.8. 電話帳グループファイル	330
15.9. マルチメディア領域	331
15.10. ブックマーク登録ファイル	331
15.11. コード認識結果ファイル	331
15.12. LED表示点灯パターン登録ファイル	333
15.13. ストレージデバイス ファイルパス	334
15.14. 内部ストレージデバイス ファイルパス	334
15.15. トルカファイル	334
15.16. ソーステンプレートファイル	335

15.17. 位置情報ファイル	336
15.18. 位置情報画像登録ファイル	336
15.19. 測位データ設定ファイル	336
15.20. スクリーンショットファイル	337
15.21. 電子コンパスデータファイル	337
15.22. サウンド設定ファイル	338
15.23. 仮想FeliCaアドホック通信設定ファイル	338
15.24. Bluetooth設定ファイル	338
15.24.1. Bluetooth通信設定ファイル	338
15.24.2. Bluetoothデバイス情報設定ファイル	340
15.24.3. 起動可能外部機器設定ファイル	340
15.24.4. MDEP設定ファイル	341
15.24.5. MDEP入力データファイル	342
15.24.6. MDEP出力データファイル	342
15.25. BML連携起動パラメータファイル	342
15.26. DTV連携起動パラメータファイル	342
15.27. 視聴/録画予約情報ファイル	343
15.28. 「アプリケーション名.properties」ファイル	344
15.29. 仮想FeliCaデータファイル	346
15.30. スクラッチパッドデータファイル	347
15.31. ハイパーマルチデータファイル	347
15.32. 加速度センサデータファイル	347
15.33. 動作認識データファイル	348
15.34. 音声認識データファイル	348
15.34.1. マスター設定ファイル	348
15.34.2. 音声取得情報設定ファイル	348
15.34.3. 認識結果情報設定ファイル	349
15.35. 対応フォントサイズ設定ファイル	350
15.36. 署名者情報ファイル	350
15.37. CSV入出力ファイル	350
15.38. 歩数計ファイル	351
15.38.1. 歩数計サマリーファイル	351
15.38.2. 歩数計データファイル	351
15.39. 歩数計設定ファイル	352
15.40. ケータイ払いデータファイル	352
15.41. ポイント払いデータファイル	352

15.42. 個別課金コンテンツ権利データファイル	353
15.43. 都市情報設定ファイル	353
15.44. コピー文字列起動パラメータファイル	353
15.45. 身体活動量計データファイル	353
15.46. オートGPS制御情報データファイル	355
15.47. ジャイロセンサデータファイル	356
15.48. 身体情報データファイル	357
15.49. 脈拍計データファイル	358
15.50. 赤外線受信データファイル	358
15.51. 音楽プレイヤー関連ファイル	358
15.51.1. 音楽コンテンツデータファイル	359
15.51.2. プレイリストファイル	362
15.51.3. レジューム情報ファイル	362
15.51.4. 画像ファイル格納領域	363
15.52. RAW画像取得機能関連ファイル	363
15.52.1. RAW画像取得設定ファイル	363
15.52.2. JPEG画像データファイル	364
15.52.3. カメラファインダ設定ファイル	364
15.53. 開発支援情報表示設定ファイル	364
15.54. ファイルの記述具体例	366
第 16 章 証明書の管理	367
第 17 章 OBEXの利用	369
第 18 章 再配布不可識別子	369
第 19 章 Eclipse の利用	370
19.1. Eclipseについて	370
19.2. 「Eclipseプラグイン for iappli」 のインストール	371
19.2.1. 始める前に	371
19.2.2. インストール	371
19.3. 新規アプリケーションの作成	372
19.3.1. プロジェクトメニューのカスタマイズ	372
19.3.2. プロジェクトの作成	373
19.3.3. ソースファイルの作成	377
19.3.4. プロジェクト設定からのソースファイルの作成	380
19.3.5. アプリケーションディスクリプタファイル (ADF) 設定	381
19.3.5.1. FeliCaCPIDの設定	381
19.3.5.2. FeliCaフリー領域の関連付け	382
19.3.6. トラステッド動作の設定	383
19.3.7. アプリケーションの構築と実行	384
19.3.8. プロジェクトの設定	387

19.3.9. ファイル配置カスタマイズ	388
19.3.10. i アプリライブラリの作成	389
19.3.11. FeliCa Push通知パラメータ設定	390
19.3.12. i アプリライブラリの利用	391
19.3.13. デバッグの実行	393
19.3.14. i アプリからDTV起動を行う場合の注意事項	399
19.4. 各種設定	400
19.4.1. 全般設定	400
19.4.2. ターゲットプロファイルの変更	401
19.4.3. ADF/トラステッド動作設定チェック処理の設定	405
19.4.4. ネットワークの設定	407
19.4.5. メールアドレスの設定	408
19.4.6. エミュレーション端末ウィンドウの設定	409
19.4.7. 開発支援情報表示設定	412
19.4.8. 電話帳の編集	413
19.4.9. グループの編集	415
19.4.10. 送受信メールの編集	416
19.4.11. ブックマーク一覧	418
19.4.12. サウンド設定	419
19.4.13. 位置情報編集	420
19.4.14. 測位データ編集	422
19.4.15. コンパス情報編集	424
19.4.16. ToruCaデータ編集	426
19.4.17. Bluetooth設定	428
19.4.18. Bluetooth HDP設定	429
19.4.19. 仮想FeliCa設定	431
19.4.20. 視聴/録画予約一覧	432
19.4.21. ハイパーマルチ機能設定	433
19.4.22. 加速度センサデータ編集	434
19.4.23. 動作認識データ編集	436
19.4.24. 音声認識データ編集	438
19.4.25. 歩数計データ編集	441
19.4.26. 身体活動量計データ編集	444
19.4.27. オートGPS制御情報一覧	447
19.4.28. ジャイロセンサデータ編集	448
19.4.29. 身体情報編集	450
19.4.30. 脈拍計データ編集	451
19.4.31. 音楽コンテンツ編集	452
19.4.32. プレイリスト編集	454
19.5. 仮想FeliCaデータファイルビューワ	456
19.5.1. 仮想FeliCaデータファイル	456
19.5.2. 仮想FeliCaデータファイルビューワの起動	457
19.5.3. 階層表示	458
19.5.4. バイナリ表示	460
19.5.5. フリー領域を扱う i アプリの利用	461
19.6. スクラッチパッド	463
19.6.1. スクラッチパッド管理ツール	463
19.6.2. スクラッチパッド管理ツールの起動	463
19.6.3. スクラッチパッドを全て初期化する	465

19.6.4. スクラッチパッドでJARファイルを扱う場合の注意事項.....	465
19.7. ADFチェック	466
19.7.1. ADFチェックツールの起動.....	466
付録.1 エミュレーション端末の設定.....	468
1. サウンドライブラリの設定.....	468
2. OpenGL/ES機能の設定.....	468
用語の説明.....	469

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、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 MascotCapsule V3、V4、eruptionST を搭載しています。MascotCapsule 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 [evo] 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」、「MyFACE」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 1997-2011 Aplix Corporation. All Rights Reserved.

Copyright 2000-2012 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 Star-2.0」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 Star-2.0」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。

「iα ppli Development Kit for Star-2.0」では、DoJa プロファイル向けの i アプリを作成することはできません。

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 Star-2.0 API、Star-2.0 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 Star-2.0 の動作環境

「iα ppli Development Kit for Star-2.0」は、以下の環境での動作を保証します。

- Windows XP SP3 / Windows 7 ※
- Visual C++ 2005 SP1 ライブラリ ランタイム コンポーネント（ATL のセキュリティ更新プログラム適用版）
- Java(TM) 2 Platform Standard Edition Development Kit 6.0 Update 22 (JDK)
- Java(TM) 2 Platform Standard Edition Runtime Environment 6.0 Update 22 (JRE)

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。ただし、JDK の 5.x 以前には非対応です。

※ステレオ表示機能において、3D 表示を利用する場合は、OS は Windows 7 のみの対応となります。さらに、上記に加え、以下の環境が必要となります。

- DirectX（February 2010 以降）
- NVIDIA 3D Vision-Ready 環境

なお、3D 表示は、全ての NVIDIA 3D Vision-Ready 環境での動作を保証するものではありません。

※また、描画最適版は以下の 2 社製のグラフィックカードを搭載した環境でのみインストールできます（インストール時、「setup.exe」により描画最適版をインストール可能かどうか自動判別します）。

- NVIDIA Corporation
- ATI Technologies, Inc.

ただし、描画最適版については、上記 2 社製のグラフィックスカードを搭載したすべての環境での動作を保証するものではありません。

1.1.1. iα ppliTool について

「iα ppli Development Kit for Star-2.0」をインストールすると「iα ppliTool」が実行できるようになります。

「iα ppliTool」の起動方法とその詳細については、「第3章.iα ppliToolの起動」で説明します。

1.2. iα ppli Development Kit for Star-2.0 の概要

「iα ppli Development Kit for Star-2.0」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。「iα ppli Development Kit for Star-2.0」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. iα ppli Development Kit for Star-2.0 の利点

「iα ppli Development Kit for Star-2.0」には、次の利点があります：

- i アプリエミュレータ機能により、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（「9.7. 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。「iα ppli Development Kit for Star-2.0」では、Eclipse 3.6 で i アプリを開発するために必要な拡張モジュール（プラグイン）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 19 章. Eclipse の利用」を参照してください。

- Eclipse : <http://www.eclipse.org/downloads/>

1.2.3. エミュレーションの範囲

ここでは、「iα ppli Development Kit for Star-2.0」のエミュレータ機能（以降、「エミュレータ」と呼びます）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：

- **「iα ppli Development Kit for Star-2.0」は Star プロファイル向けのため、DoJa プロファイル向けの i アプリを作成、および実行する事はできません。**
- ADF のサイズに制限はありません。またアプリケーションのサイズにも制限はありません。ただし、i アプリの規定上限サイズ（JAR ファイルのサイズがミニアプリの場合は 50KByte、フルアプリ、フル+ミニ、ミニ+フルアプリの場合はスクラッチパッドサイズと合わせて 10,240KByte）を超える場合、起動前に警告メッセージが表示されます。
- 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 `StarApplication.LAUNCHED_AFTER_DOWNLOAD` を発生させるテストが可能です。
- アプリケーション起動時画像を表示する機能はありません。ただし、画像確認画面より起動時画像をプレビューすることが可能です。
- 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 `StarApplication.LAUNCHED_FROM_TIMER` を発生させるテストが可能です。

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「4.8. 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：

- バックライトやパイプレータの状態はエミュレーション端末画面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。
- テレビ電話機能はありません。
- マナーモード状態の設定は可能ですが、マナーモード中の音声出力可否設定が音声出力不可の状態でも音声出力されます。

4. 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、メーラ、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 `StarApplication.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。
- メーラから 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。
- 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機能で赤外線信号を発信する事はできません。
- 実際に動画再生、音楽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 で位置情報を測位したり、オート GPS サーバへ位置情報データを送信することはできません。
- 実際に PKI の PKCS#7 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。またシンプルペアリング接続時には実際に FeliCa は使用しません。
- 実際に DTV を視聴する事はできません。
- 実際に音声認識の DDP 通信はできません。
- 実際に招集メッセージを送受信する事はできません。
- 実際に個別課金サーバへ接続したり、コンテンツをダウンロードする事はできません。また実際にケータイ払い決済やドコモポイント交換は行えません。
- 実際に音声合成機能による音声出力はできません。

5. 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。

6. 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。

本版における制限事項については、別途、「Star-2.0 API i α ppli Development Kit リリースノート」を参照してください。

1.3. iα ppli Development Kit for Star-2.0 によるiアプリの開発

ここでは、「iα ppli Development Kit for Star-2.0」による i アプリの開発サイクルについて説明します。

「iα ppli Development Kit for Star-2.0」を使用した場合、i アプリの開発サイクルは、次のような流れになります：

1. i アプリのプロジェクト作成
「iα ppliTool」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
「iα ppliTool」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
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。
5. プロジェクトのビルド
「iα ppliTool」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、「iα ppliTool」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
JDK 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 Star 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。
 - パッケージング
JDK の jar ツールを使用して、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）の更新
アプリケーションディスクリプタファイルの、JAR ファイルサイズ、日付に関する記述を更新します。
 - 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. i アプリを実行
「iα ppliTool」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。

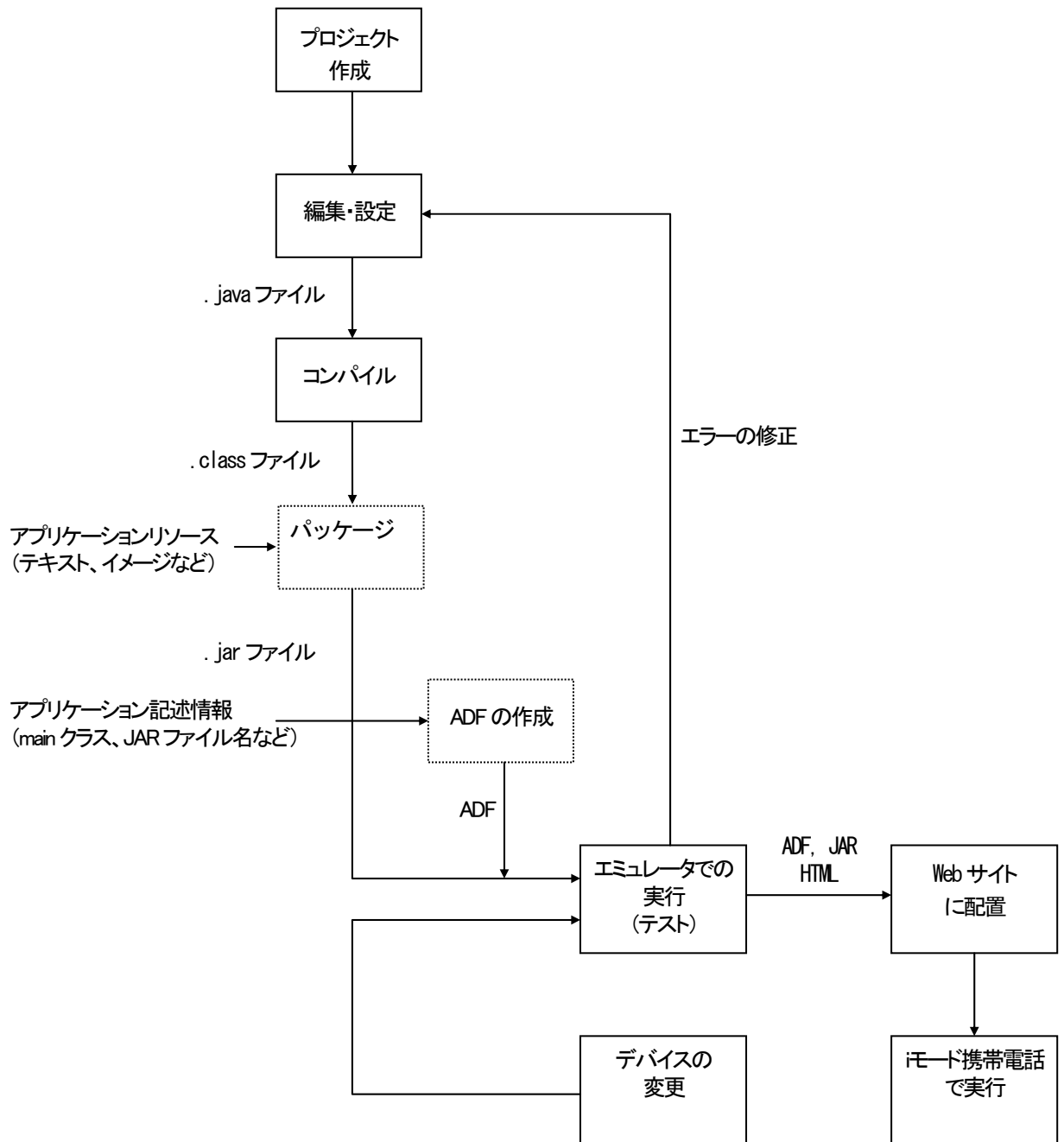


図 1-1 アプリケーション開発サイクル

第2章 インストール

この章では、「iα ppli Development Kit for Star-2.0」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

「iα ppli Development Kit for Star-2.0」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 「iα ppli Development Kit for Star-2.0」を動作させるには、**JavaTM 2 Platform Standard Edition Development Kit (JDK)** がインストールされている必要があります。
もし、JDK がインストールされていない場合は、Oracle のサイトから JDK を入手してインストールしてください。
- ② 「iα ppli Development Kit for Star-2.0」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。
- ③ Eclipse 統合開発環境を使う場合で、他のプロファイル（DoJa-5.1 や Star-1.2 など）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_Star-2.0)

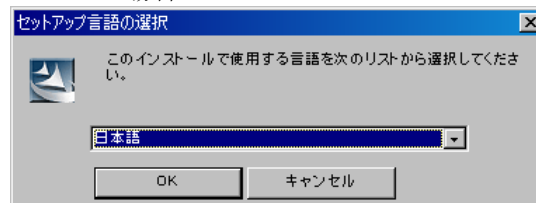
2.2. インストール手順

「iα ppli Development Kit for Star-2.0」のインストールは、管理者権限を持つユーザでログインした状態で行ってください。

「iα ppli Development Kit for Star-2.0」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：

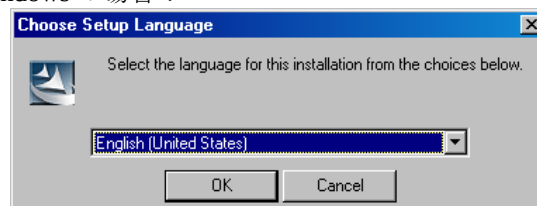


図 2-1 セットアップ言語の選択画面

3. ウィザードに従い「次へ」を押します。

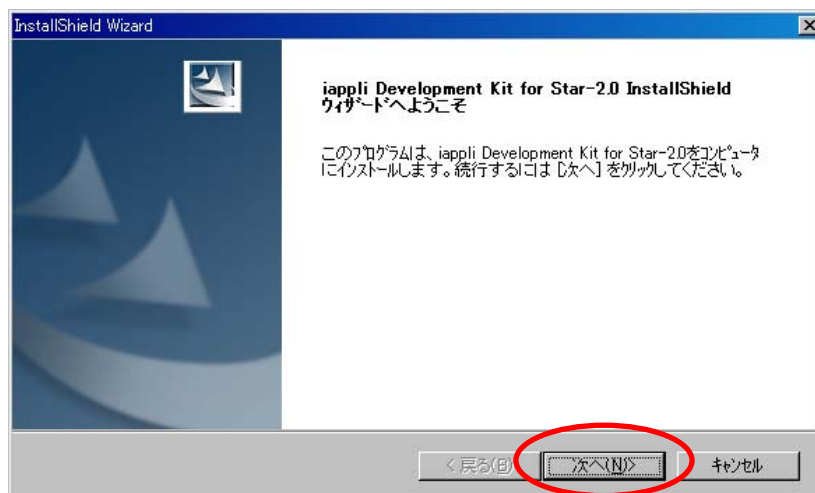


図 2-2 ようこそ画面

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。

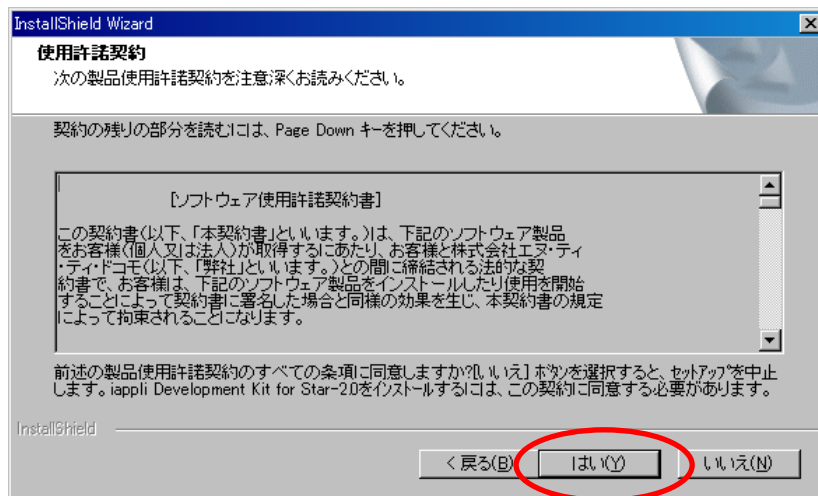


図 2-3 使用許諾契約画面

5. 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。

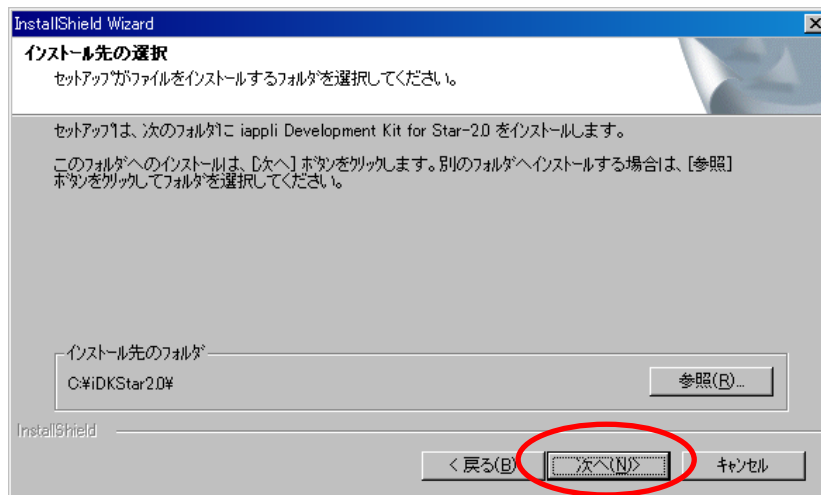


図 2-4 インストール先の選択画面

^[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。

6. 描画最適版をインストール可能な環境の場合、インストールするプログラムの種類を選択して、「次へ」を押します（描画最適版のインストールに適さない環境の場合、この画面は表示されずに手順 7 へ進みます）。

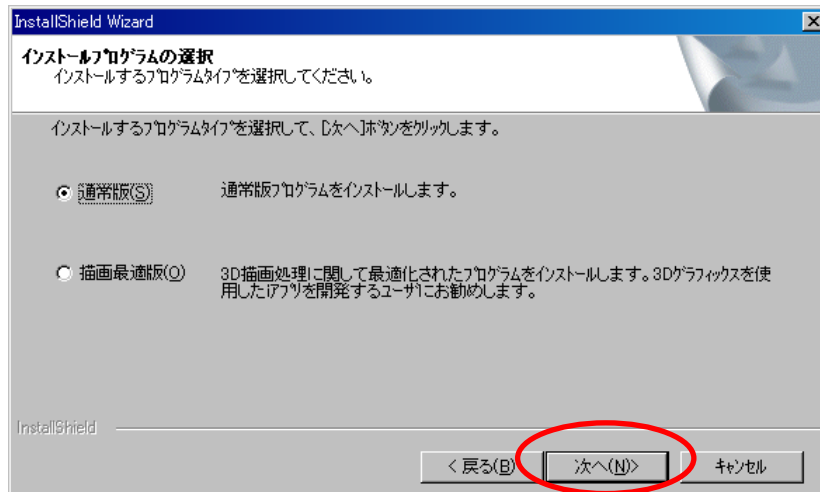


図 2-5 インストールプログラムの選択画面

- 通常版 : 通常版のプログラムをインストールします。
- 描画最適版 : 3D 描画処理に関して最適化されたプログラムをインストールします。

7. セットアップの種類を選択して、「次へ」を押します。

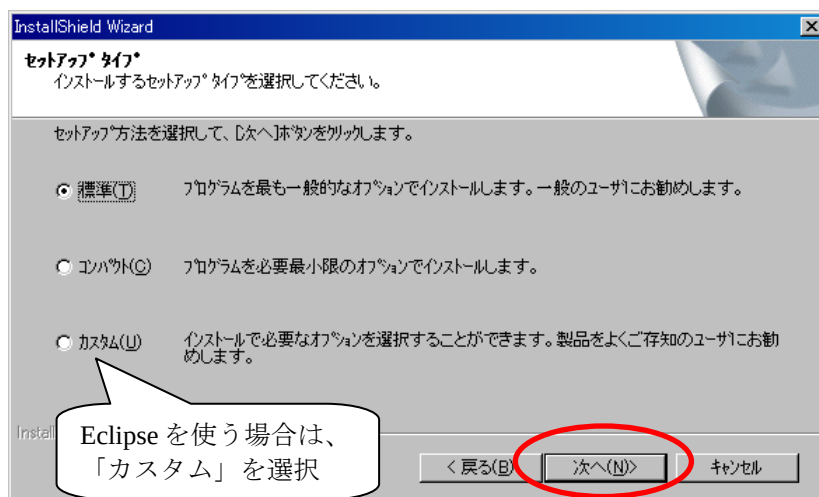


図 2-6 セットアップタイプの選択画面

- 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、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。

8. 手順7で、「カスタム」以外を選択した場合は、手順10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。

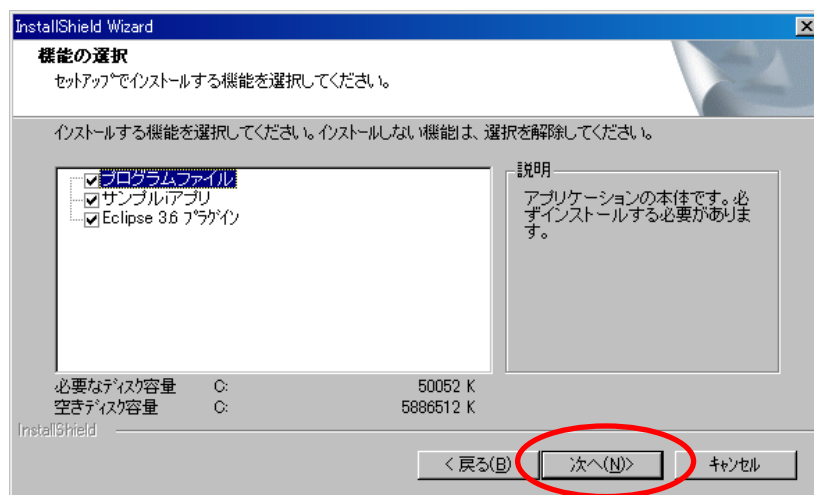


図 2-7 機能の選択画面

- 「Eclipse 3.6 プラグイン」を選択すると、Eclipse 3.6 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。

Star-2.0 以前のプラグインがインストールされている場合に、Star-2.0 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「Eclipse 3.6 プラグイン」を選択した場合、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ので、インストール先を指定してください。

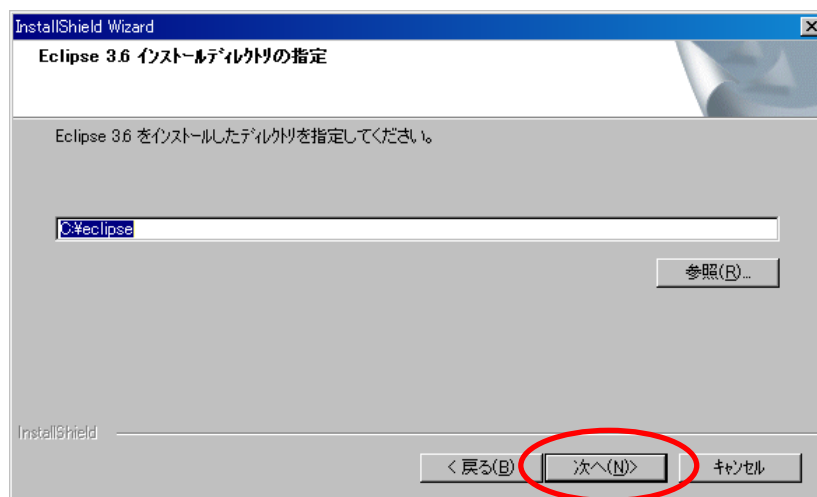


図 2-8 Eclipse プラグインインストール先設定画面

10. 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。

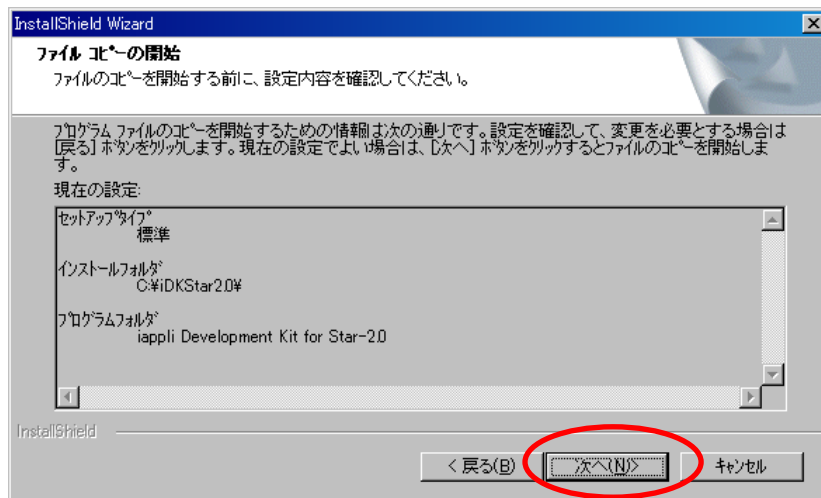


図 2-9 ファイルのコピー開始画面

11. Windowsにプロキシサーバ情報が設定されている場合、「iα ppliTool」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。プロキシサーバ情報の設定は、後からでも変更できます（「6.1.1 プロキシサーバの設定」参照）。
- Windows のプロキシサーバの情報を利用している場合：

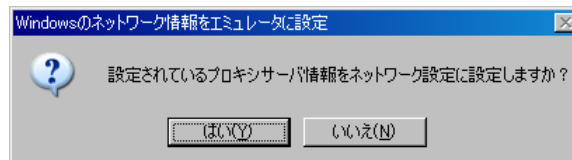


図 2-10 プロキシ情報の設定画面

- Windows のプロキシサーバの情報を利用していない場合：

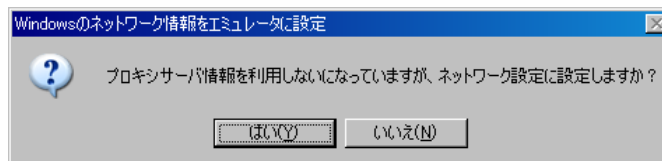


図 2-11 プロキシ情報の設定画面

12. i アプリ（.jam ファイル）を「iα ppliTool」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「iα ppliTool」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、「iα ppliTool」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（「8.12. .jam に関連付ける」参照）。

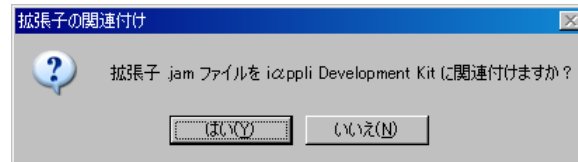


図 2-12 .jam ファイルの関連付け設定画面

13. Microsoft Visual C++ 2005 再頒布可能パッケージをインストールするかどうかを問い合わせるので、もし、Microsoft Visual C++ 2005 SP1 ランタイムライブラリをインストールしていない場合は、「はい」を押します。

このライブラリがインストールされていないと「iα ppliTool」を起動することはできません。もし、お使いのマシンにインストールされていない場合は、必ず「はい」を選択してください。
また、既にインストールされている場合でも、ライブラリのバージョンが古いと「iα ppliTool」が正常に動作しない場合があります。特に「iα ppli Development Kit for Star-1.3」までの「iα ppli Development Kit」に同梱されていた Microsoft Visual C++ 2005 SP1 再頒布可能パッケージを「iα ppli Development Kit for Star-2.0」で使用することはできません。



図 2-13 VC++ 2005 SP1 再頒布可能パッケージの設定画面

※ お使いのパソコンが x86 系マシン以外の場合は「いいえ」を選択し、別途、Microsoft のホームページから対応する最新の再頒布可能パッケージを入手してください。

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。

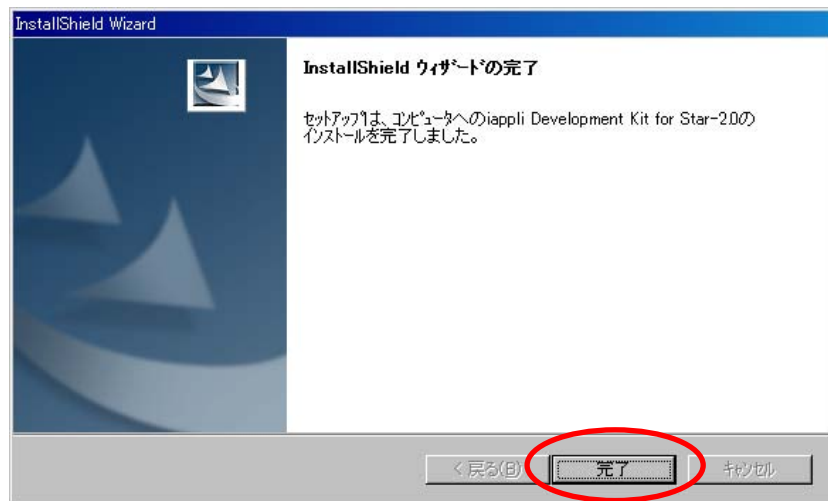


図 2-14 インストール完了画面

15. 日本語版Windowsで「i appli Development Kit for Star-2.0」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 2-15）。

もし、後から登録したい場合は、「i appli Development Kit for Star-2.0」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。

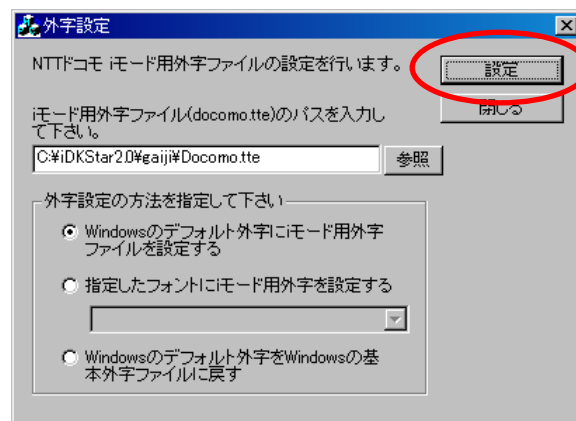


図 2-15 「ChangeEUDC.exe」の実行（外字設定画面）

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

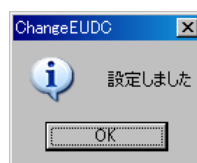


図 2-16 絵文字（外字）設定終了の確認

17. 外字設定画面の「閉じる」を押します（図 2-15）。

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 2-15 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、インストールしたディレクトリの bin の下（インストールしたディレクトリが「C:\¥iDKStar2.0」の場合、「C:\¥iDKStar2.0¥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事ができます。

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:\¥iDKStar2.0」の場合、「C:\¥iDKStar2.0¥doc」の下です。）

本書では、

「i α ppli Development Kit for Star-2.0」を「C:\¥iDKStar2.0」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、「i α ppli Development Kit for Star-2.0」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（「8.4. 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

「iα ppli Development Kit for Star-2.0」のアンインストールは、管理者権限を持つユーザでログインした状態で行ってください。

「iα ppli Development Kit for Star-2.0」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』→『iα ppli Development Kit for Star-2.0』→『Remove iα ppli Development Kit for Star-2.0』を実行します（「アプリケーションの追加と削除」から削除することもできます）。
2. 削除の確認画面が表示されるので、「OK」を押します（図 2-17）。

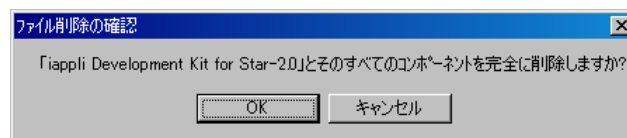


図 2-17 ファイル削除の確認画面

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:¥iDKStar2.0」を直接削除してください。

第3章 iαppliTool の起動

「iαppliTool」を起動するには、Windowsの『スタート』→『プログラム』→『iαppli Development Kit for Star-2.0』→『iαppliTool for Star-2.0』を実行します。^{[2][3]}

※Windows 7環境において、ファイルの関連付けの設定・解除操作（「8.12 .jam に関連付ける」参照）を行うためには、「iαppliTool」を管理者モードで起動する必要があります。

『スタート』→『プログラム』→『iαppli Development Kit for Star-2.0』→『iαppliTool for Star-2.0』を右クリックし、『管理者として起動...』を選択してください。その後、「ユーザー アカウント制御」ダイアログが表示される場合は「はい」を選択してください。

「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、「iαppliTool」画面であり、i アプリのビルドや実行、携帯電話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。

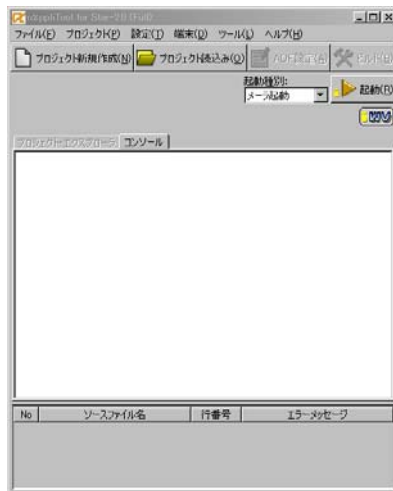


図 3-2 「iαppliTool」画面

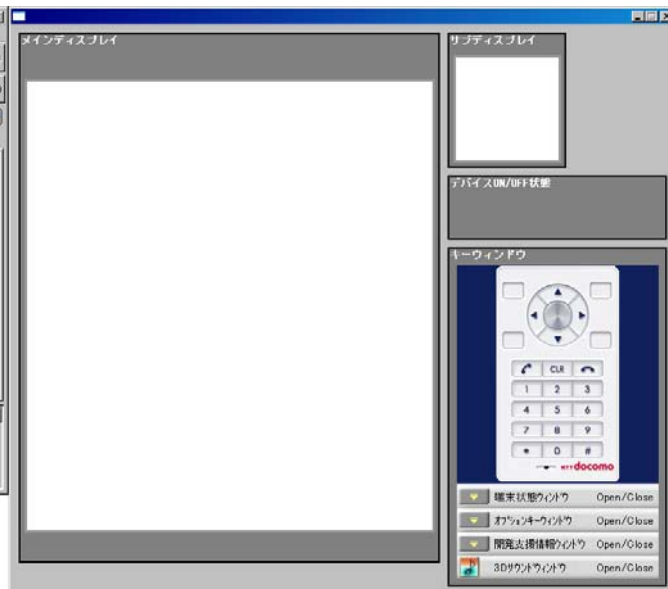


図 3-1 エミュレーション端末

^[2] 「iαppliTool」の起動時「コンピューターに MSVCR71.dll がないため、プログラムを開始できません」といった内容のメッセージが表示され起動に失敗する場合があります。これは「iαppliTool」が使用する JDK6.0 が MSVCR71.dll を使用しているためです。上記のメッセージが表示された場合は、JDK のインストールフォルダ¥bin にある MSVCR71.dll を、C:¥iDKStar2.0¥bin にコピーしてください。

^[3] 「iαppliTool」でプロジェクトのビルド（「4.4. プロジェクトのビルドと通常起動」参照）を実行したとき、「iαppliTool」が JDK 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 JDK 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 JDK 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（「4.3.3. 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。

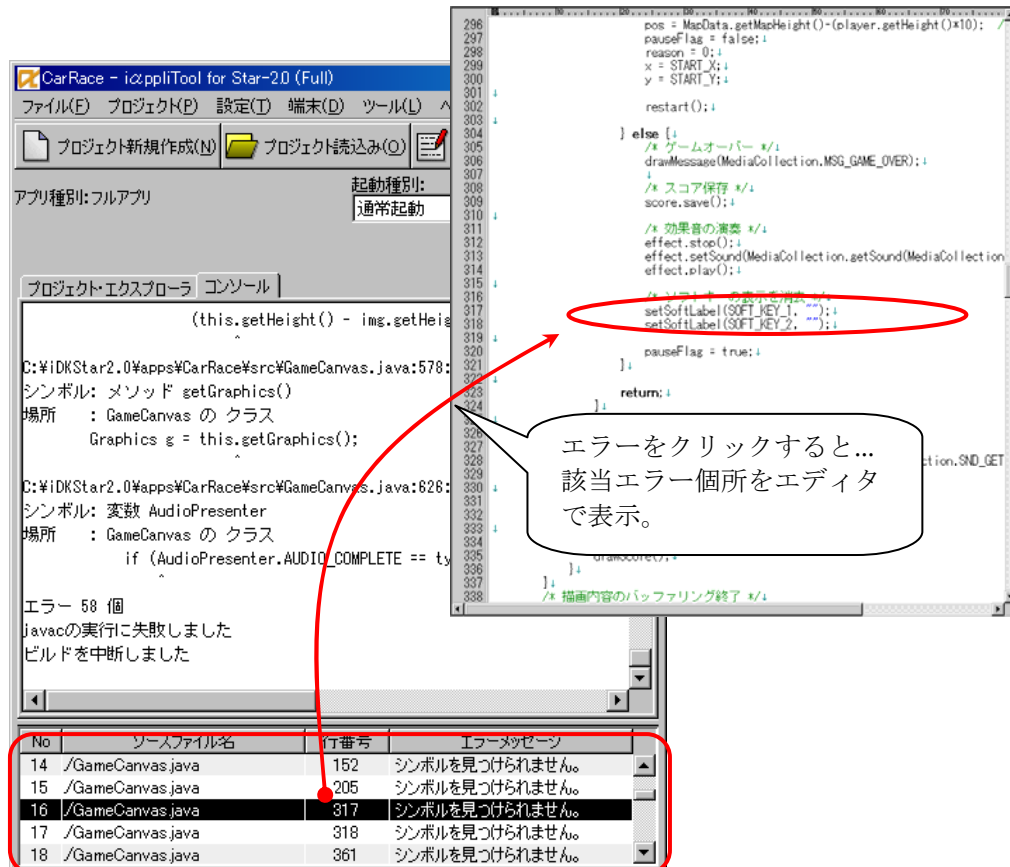


図 3-3 ビルドエラービュー

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（「4.3.3. 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（「4.8. 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（「4.6. 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（「4.9. 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（「9.1. 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（「9.3. 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（「9.7. 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。
- 右クリックの「コピー」&「貼り付け」もしくは、Windowsのファイルエクスプローラからドラッグ&ドロップの操作でファイルをコピーできます。
- 右クリックの「ソースファイルの新規作成」で、ソースファイルを新たに作成する事ができます。（この操作は、**src** フォルダ配下で可能です。）
- ミニ+フル、フル+ミニアプリにおいて、構成されているフルアプリ、ミニアプリのソースや **.jam** ファイル等を変更する事ができます。

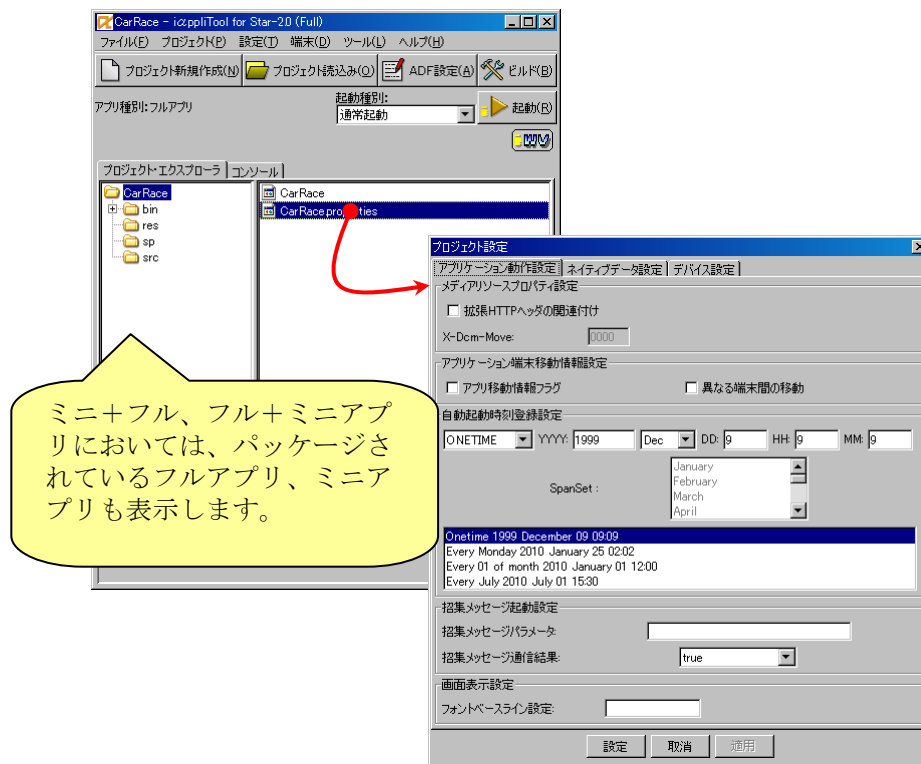


図 3-4 プロジェクト・エクスプローラ



図 3-5 ポップアップメニュー

第4章 iアプリの作成

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、

DoJa 向けで、

- 通常実行する [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](#)

Star 向けで、

- 通常実行する [フルアプリ](#)
- Widget View 上で実行する [ミニアプリ](#)
- それら両方を持っている [ミニ+フル、フル+ミニアプリ](#)
- 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「4.9. トラステッド動作の設定」を参照してください）。

「iα ppli Development Kit for Star-2.0」では、Star 向けのアプリケーションを作成する事ができます。

(DoJa 向けのアプリケーションには対応しておりません。)

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ようになります：

- 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「iα ppli Development Kit for Star-2.0」で簡単に作成することができます。

4.1. フルアプリとミニアプリ

Star では、DoJa と同じく通常実行するフルアプリ（ADF の AppType が FullApp）と、待ち受けアプリケーションに代わって、Widget View で動作するミニアプリ（ADF の AppType が MiniApp）を作成することができます。

ミニアプリは、通常の i アプリよりも機能が制限された i アプリで、待受け画面上で 1 つしか起動できなかった DoJa の待ち受けアプリケーションに代わり、Widget View 上で最大 9 個まで同時起動することができます。

Widget View とは、Star 対応 i モード携帯電話で新たに採用された機能で、ミニアプリの実行環境です。待受け画面と切り替えて表示でき、そのサイズは、WVGA(480×800 dots) です。

Widget View は、複数のミニアプリが起動している「一覧表示」、起動するミニアプリを追加する「ランチャー表示」、1 つのミニアプリを選択表示させる「個別表示」の 3 つの表示形態を持っており、ユーザ操作で切り替えられます（図 4-1）。

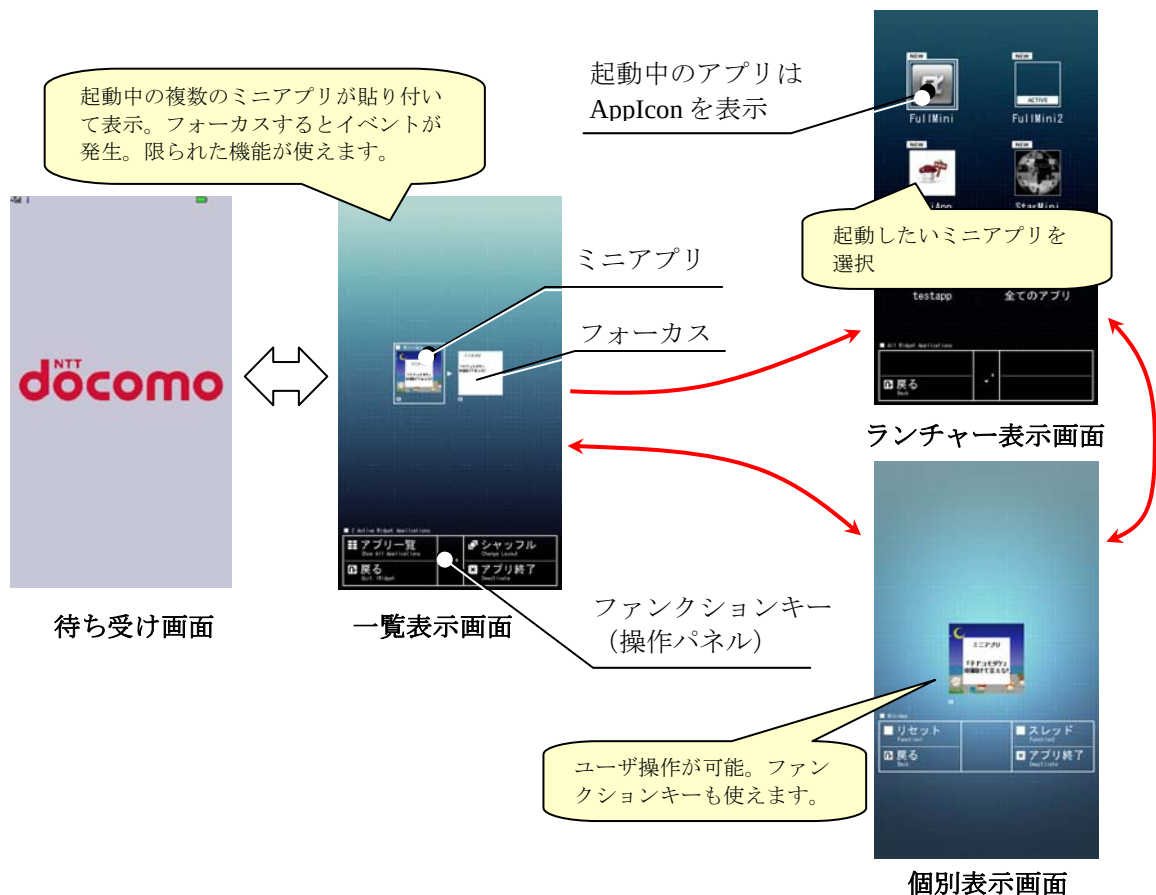


図 4-1 ミニアプリと Widget View の概要

このとき、ミニアプリは、Widget View ではない「非表示状態」、一覧表示でフォーカスを取得している「**Focused 状態**」、一覧表示でフォーカスを取得していない「**Unfocused 状態**」、個別表示で選択されている「**Selected 状態**」、個別表示で選択されていない「**Unselected 状態**」の 5 つの表示状態と、中断中の「**Suspended 状態**」、終了している「**Terminated 状態**」、

実行中の「**Started 状態**」、実行中で制限なしの「**Full-Active 状態**」、実行中で制限があり省電力の「**Semi-Active 状態**」の5つの実行状態を持っています。

各状態の関係は、下記、表 4-1 に示すとおりです。

ミニアプリにおける状態の詳細については、別途、API リファレンス等を参照してください。

表 4-1 Widget View におけるミニアプリの状態

Widget View の表示状態	ミニアプリの表示状態	ミニアプリの状態
非表示	非表示状態	Suspended
		Terminated
一覧表示	Focused 状態	Started Semi-Active Suspended
	Unfocused 状態	Started Semi-Active Suspended
	非表示状態	Terminated
個別表示	Selected 状態	Started Full-Active Suspended
	Unselected 状態	Suspended
		Terminated
ランチャー表示	非表示状態	Terminated
	非表示状態	Suspended

4.1.1. ミニアプリの画面表示構成

ミニアプリは、Widget View上のどこにでも配置でき、パレットと呼ばれる領域を設定して描画されます（図 4-2）。

パレットの値は、ADFのPallet キーにて指定します（4.8. 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）。

【注意】

エミュレータでは、一覧表示状態において、ミニアプリを自由に配置させることはできません。

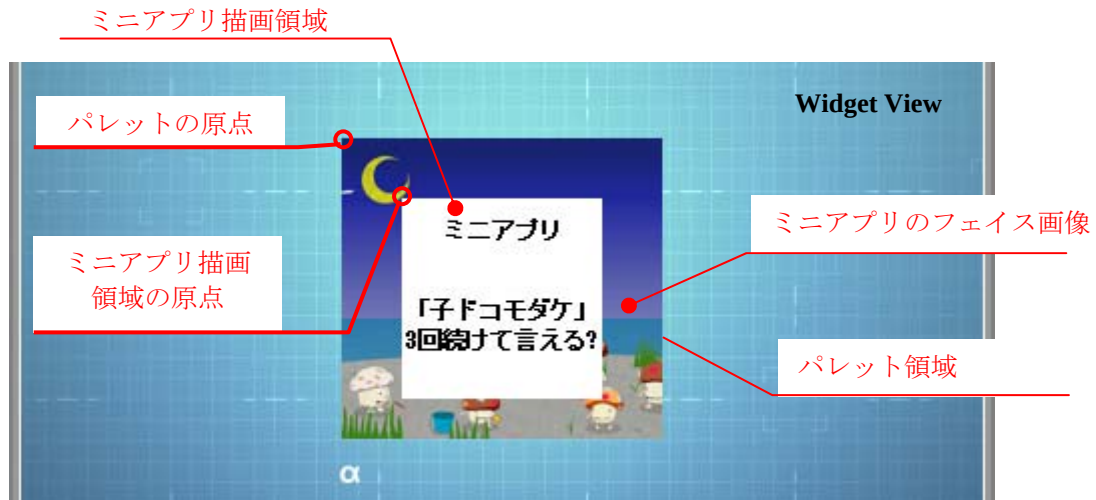


図 4-2 ミニアプリ描画におけるパレットの構成 (Pallet キーの指定)

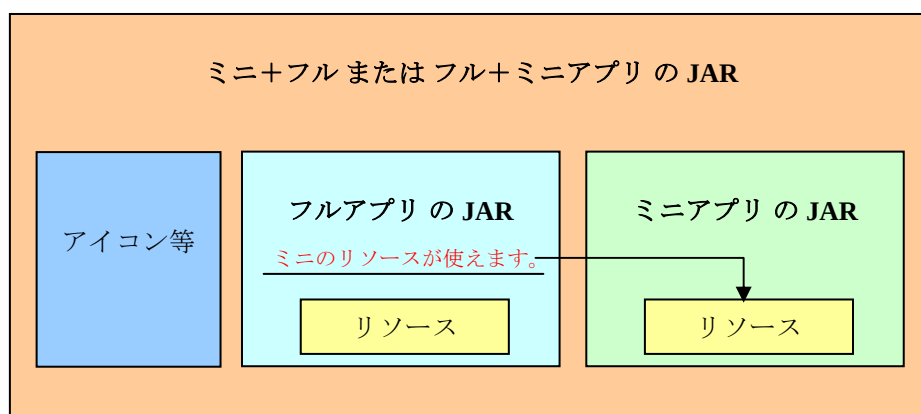
4.1.2. フル+ミニ、ミニ+フルアプリ

フルアプリとミニアプリを1つのJARファイルにパッケージしたもので、ADFのAppType、AppClass、InnerPackageName、InnerAppSize キーにて指定します（4.8. アプリケーションディレクトリファイル（ADF）の編集 参照）。

- AppType が FullApp,MiniApp の場合、フル+ミニアプリであり、優先的にフルアプリを起動します。
- AppType が MiniApp,FullApp の場合、ミニ+フルアプリであり、優先的にミニアプリを起動します。

フル+ミニアプリ、ミニ+フルアプリにおいて、パッケージされたフルアプリとミニアプリは、動作中にお互いを切り替えることができます（「第 13 章. フルミニ連携」参照）。また、フルアプリからミニアプリのリソースを扱う事ができます。

（ミニアプリからフルアプリのリソースは扱えません。）



注意：

ミニ+フル、フル+ミニプロジェクトの直下のリソース(res 配下)は、アプリケーションから扱う事はできません。主に ADF の AppIcon で指定する表示用アイコンを置きます。

4.2. iアプリの作成から起動まで

「iα ppliTool」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。

「iα ppliTool」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（「4.3. プロジェクトの新規作成」を参照）
- ② iアプリの実行（「4.4. 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（「4.5. プロジェクトの読み込み」を参照）
- ④ プロジェクトの設定（「4.6. 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（「4.8. 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。

「iα ppli Development Kit for Star-2.0」に搭載されたサンプルでも、この手法を用いています。

スクラッチパッドおよび、JARファイルサイズの制限については、「4.8. アプリケーションディスクリプタファイル（ADF）の編集 SPsize、AppSize」を参照してください。

4.3. 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 4-3）。



図 4-3 プロジェクト新規作成画面

2. プロジェクトの格納先（C:\iDKStar2.0\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ます。
4. アプリケーション種別を選択します。

ここで、「ミニ+フルアプリ」を選択した場合は、パッケージするフルアプリとミニアプリを選択する画面が表示されるので、フルアプリとミニアプリを選択後、「設定」を押すとプロジェクトが作成されます（図 4-4）。

ミニ+フルアプリ、フル+ミニアプリのプロジェクトでは必ずプロジェクト格納先に、パッケージしたいフルアプリとミニアプリのプロジェクトが必要です。

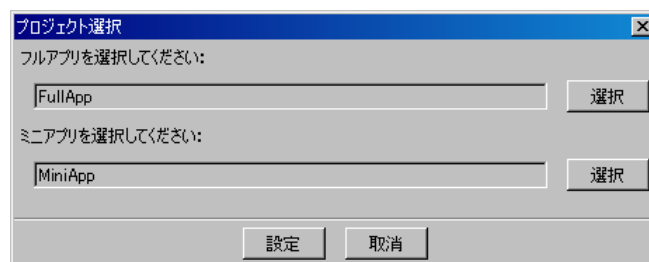


図 4-4 プロジェクト選択画面

5. 「作成」ボタンを押します。^[4]

「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の作成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 4-5）。

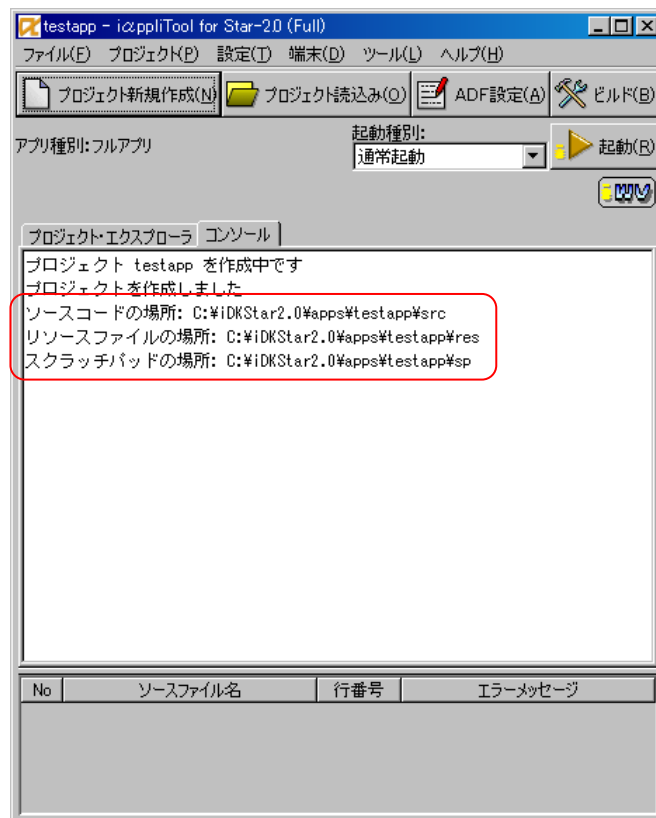


図 4-5 新規プロジェクト作成後のメイン画面

6. 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「4.3.1. ソースファイルの作成」を参照してください。

^[4] 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:\¥iDKStar2.0¥apps¥<プロジェクト名>¥src
 リソースファイルの格納場所: C:\¥iDKStar2.0¥apps¥<プロジェクト名>¥res
 （ミニ+フル、フル+ミニアプリにおいては、主にアイコン表示用であり、iアプリから扱うことはできません。）
 スクラッチパッドの格納場所: C:\¥iDKStar2.0¥apps¥<プロジェクト名>¥sp
 JAR、ADF、HTML の格納場所: C:\¥iDKStar2.0¥apps¥<プロジェクト名>¥bin

4.3.1. 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 4-6）。前述のプロジェクトを作成する手順からでも同様です。



【フルアプリの場合】

【ミニアプリの場合】

図 4-6 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:\¥iDKStar2.0¥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、
- 「フルアプリ」または「ミニアプリ」
 - 「なし」
- のいずれかを選択します。^[5]

「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。

- (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。
ただし、ミニアプリでは「Canvas」しか使えないので選択できません。



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

- (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。

- ファンクションキーリスナ・・・ソフトキーを扱う場合

^[5] 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:¥iDKStar2.0¥lib¥template¥StarApplicationPanel.txt
- Canvas 用通常アプリケーション：
C:¥iDKStar2.0¥lib¥template¥StarApplicationCanvas.txt
- Panel のみ：
C:¥iDKStar2.0¥lib¥template¥StarPanel.txt
- Canvas のみ：
C:¥iDKStar2.0¥lib¥template¥StarCanvas.txt
- 各リスナおよびイベントの設定ファイル（埋め込みの定義）：
C:¥iDKStar2.0¥lib¥template¥listener¥xxxListener.conf
C:¥iDKStar2.0¥lib¥template¥listener¥xxxEvent.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。

- 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 タイマーを扱う場合
- オーディオ 3D リスナ 3D 音源を扱う場合
- Star イベントリスナ Star イベントを扱う場合

(4) 「Star イベントリスナ」にチェックを付けると、Star イベントが選択できるようになります。

使用するイベントで、リスナに登録したいイベントにチェックを付けます。

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。

（ミニ+フル、フル+ミニアプリのプロジェクトにおいて、プロジェクト直下に置かれたリソースファイル（プロジェクト名/**res**）は、主にアイコン表示用であり、i アプリから扱うことはできません。i アプリから扱うリソースはそれぞれパッケージされるフルアプリ、ミニアプリのプロジェクトに置いてください。）

i アプリのJARファイルに含めてしまうもの：

- ・リソースファイル：**C:¥iDKStar2.0¥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:¥iDKStar2.0¥lib¥multimedia** の下

（詳しくは、「15.9. マルチメディア領域」参照）

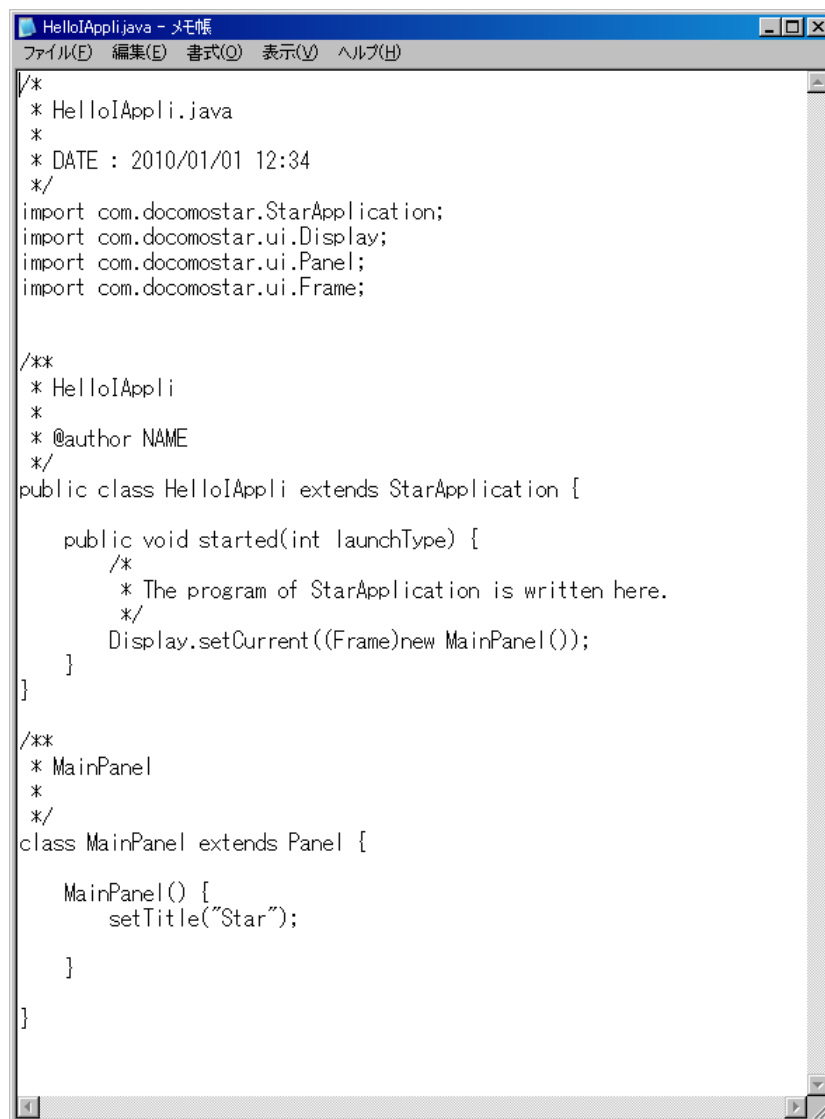
7. 任意のエディタでソースファイルが開かれるので、ソースコードを作成してください。
（任意のエディタについては、「4.3.3. ソースエディタの設定」参照）

（例）

図 4-6 でソースファイル名に“HelloIAppli”と入力し、テンプレートに「フルアプリ」の「Panel」を選択して作成すると、図 4-7 のようにファイルが開かれます。

このソースコードは、

C:¥iDKStar2.0¥apps¥<プロジェクト名>¥src¥HelloIAppli.java
で作成されています。



```
/*
 * HelloIAppli.java
 *
 * DATE : 2010/01/01 12:34
 */
import com.docomostar.StarApplication;
import com.docomostar.ui.Display;
import com.docomostar.ui.Panel;
import com.docomostar.ui.Frame;

/**
 * HelloIAppli
 *
 * @author NAME
 */
public class HelloIAppli extends StarApplication {

    public void started(int launchType) {
        /*
         * The program of StarApplication is written here.
         */
        Display.setCurrent((Frame)new MainPanel());
    }
}

/**
 * MainPanel
 *
 */
class MainPanel extends Panel {

    MainPanel() {
        setTitle("Star");
    }
}
```

図 4-7 (例) HelloIAppli ソースコード

※ 1

ソースコードを記述するには、Star の API 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 Star の API リファレンスは、
NTT ドコモから適宜入手してください。

4.3.2. 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（「4.4. 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 4-8）。

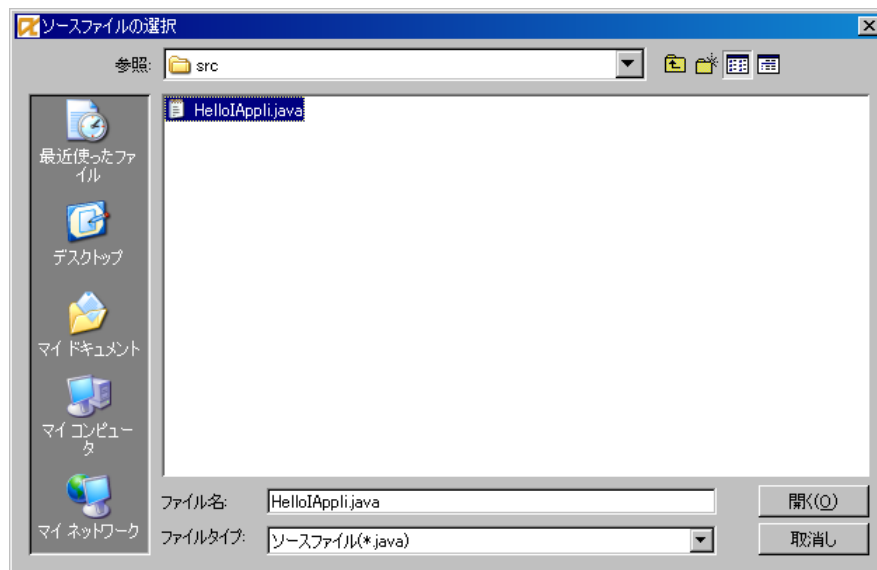


図 4-8 ソースファイルの選択画面

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、ソースコードを修正してください
（任意のエディタについては、「4.3.3. ソースエディタの設定」 参照）。

ワンポイント

この操作は、プロジェクト・エクスプローラからソースファイルをダブルクリックすることでもできます（「3.2. プロジェクト・エクスプローラ」 参照）。

4.3.3. 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「メモ帳（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 4-9）

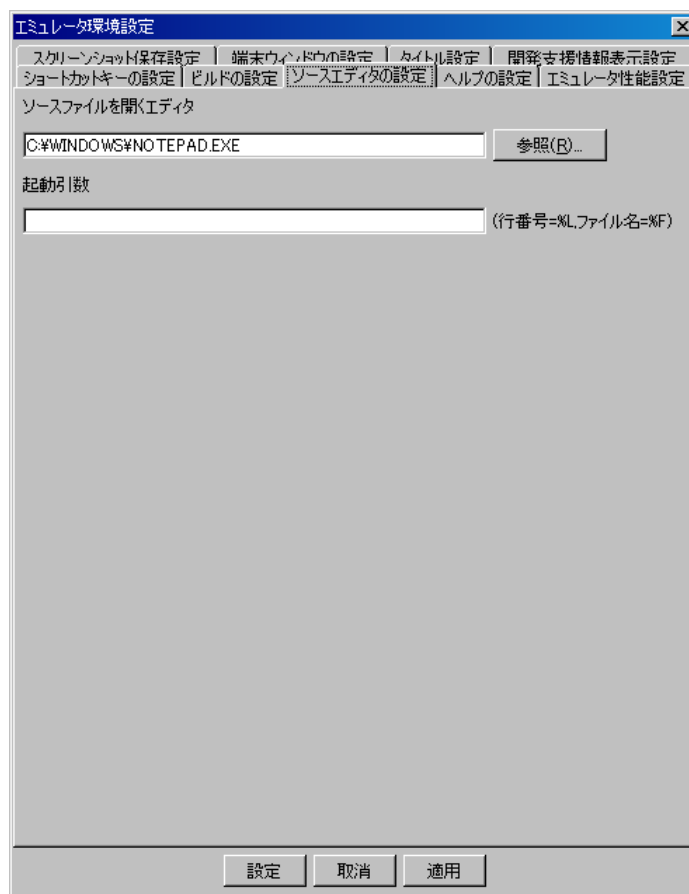


図 4-9 ソースエディタの設定画面

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。

4.4. プロジェクトのビルドと通常起動

DoJa プロファイル向け i アプリのビルドは実行できません。

「iα ppli Development Kit for Star-2.0」でビルドした i アプリ (JAR ファイル) は、Star-1.3 プロファイル搭載以降の携帯電話でのみ実行可能です。それ以前の携帯電話では実行できませんのでご注意ください。

(「iα ppli Development Kit for Star-2.0」のビルド機能によってJARファイルを生成すると、プロジェクト読み込み画面に  アイコンが表示されます。詳しくは「4.5.1 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」を参照ください)

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 4-10)。

ソースコードのコンパイルを行い、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。^[6]

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StarApplication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、「4.8. 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。

^[6] 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 JDK 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 JDK をインストールしたディレクトリを設定した後、改めて「iα ppliTool」を起動してください。

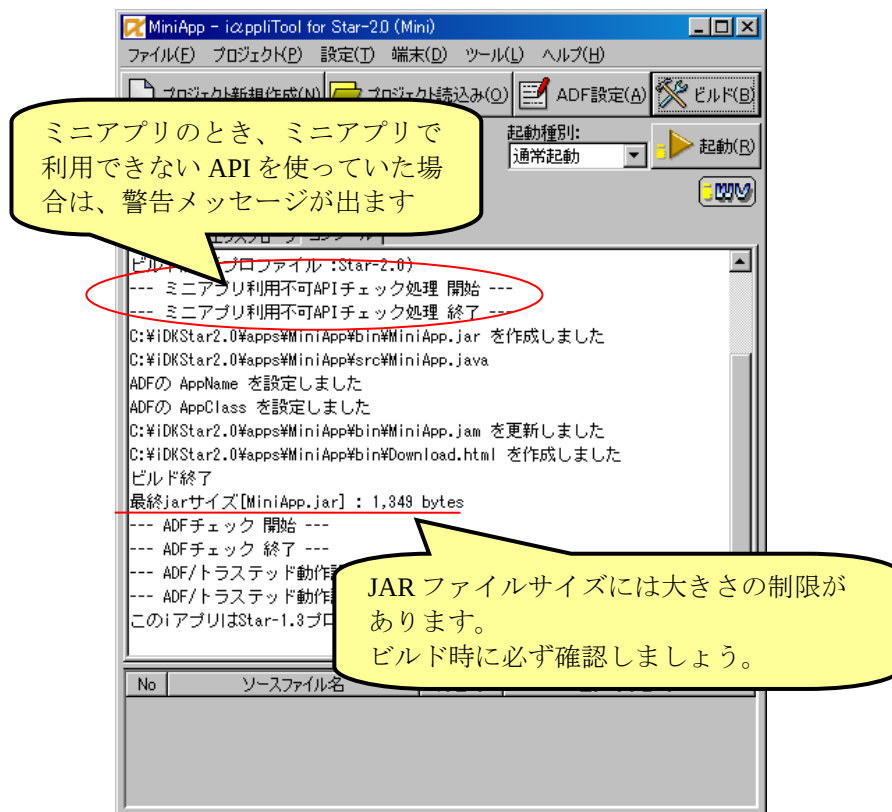


図 4-10 プロジェクトのビルド

- 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 4-11）。
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（「4.8. アプリケーションディスク립タファイル（ADF）の編集」参照）。

ADF設定

フィルター:

AppType FullApp

AppName HelloIAppli

AppVer

PackageURL testapp.jar

AppSize 922

ConfigurationVer

ProfileVer

SPsize

AppClass HelloIAppli

AppParam

LastModified Fri, 1 Jan 2010 12:34:56

UseNetwork ☐ yes

TargetDevice

LaunchAt

UseTelephone ☐ call

UseBrowser ☐ launch

UseDTV ☐ launch

設定 取消 画像確認

図 4-11 ADF の設定

(図 4-7 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 4-13)。

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」、「4.34 MyFACE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 (「4.23. トルカ起動」を参照) が起動している。
- BMLブラウザ (「4.25. BML起動」を参照) が起動している。
- DTV (「4.26. 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。起動タイプを変更する場合はメイン画面の「起動種別」プルダウンで起動タイプを選択し起動します。(起動方法と起動タイプについては「4.10 受信フォルダ起動」以降に説明しています)

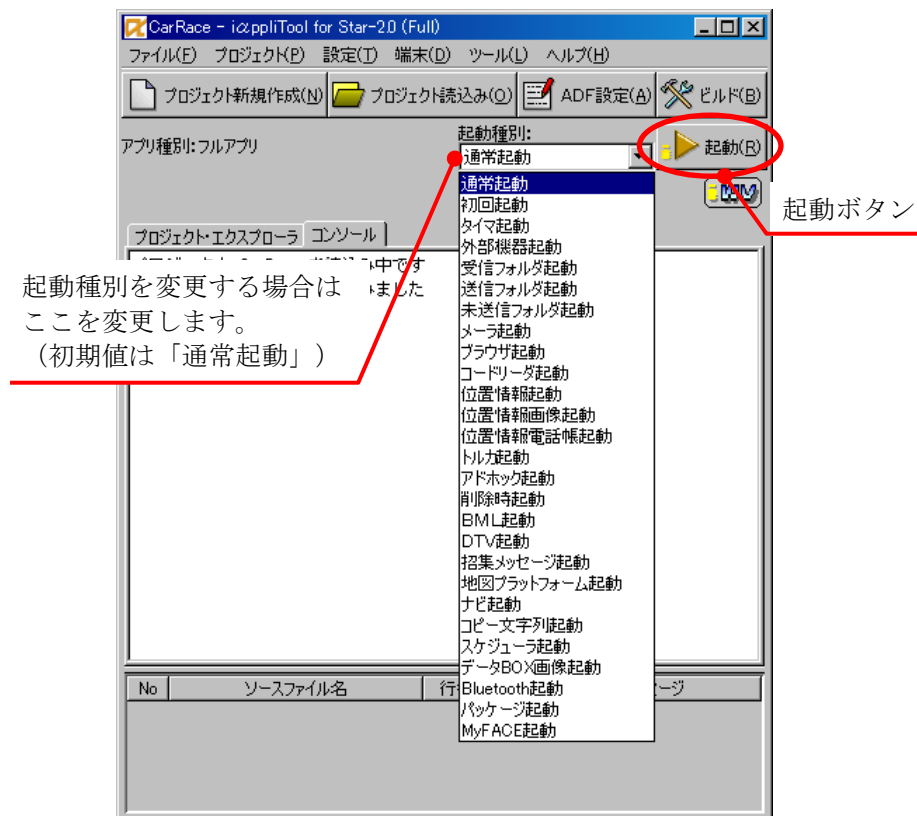


図 4-12 起動種別の選択

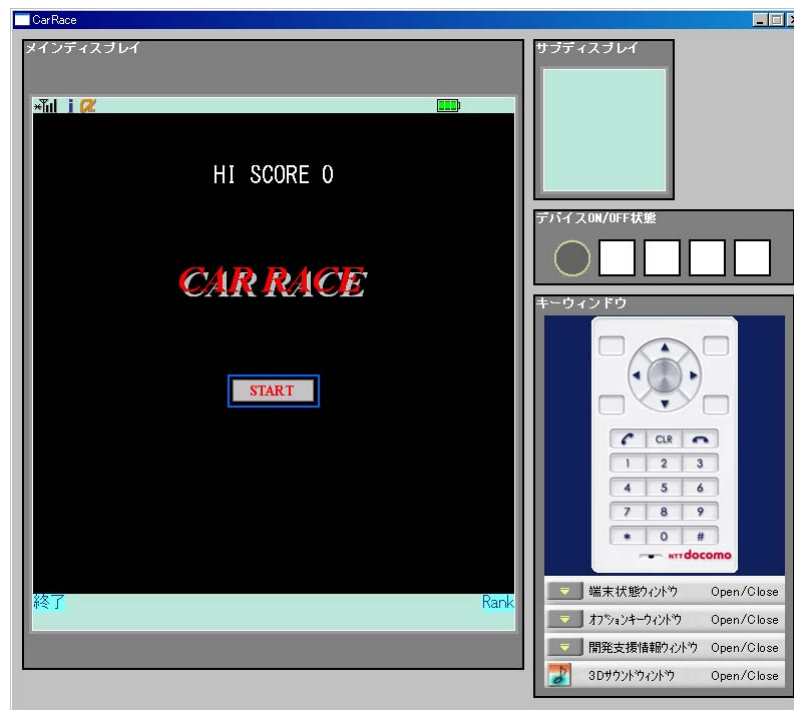


図 4-13 i アプリ 起動

4.4.1. 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 4-14）。

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。

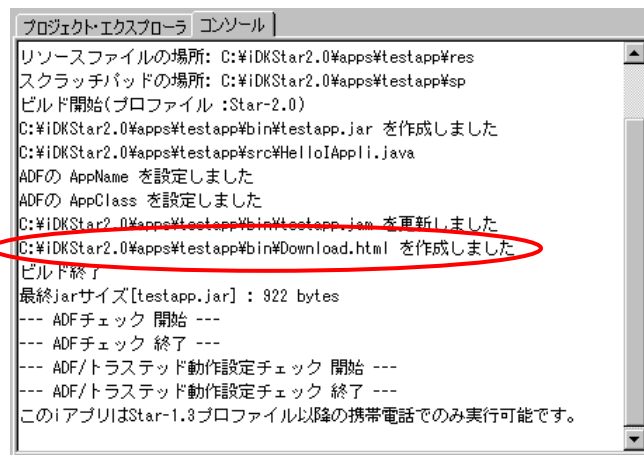


図 4-14 ダウンロード HTML ファイル作成

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:\¥iDKStar2.0¥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 (4.8. 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、ista のいずれかになります。 ijam・・・ブラウザ起動、MyFACE 起動以外の場合 ista・・・ブラウザ起動、MyFACE 起動の場合

HTMLファイルのタグ記述については、「15.4. ブラウザ起動HTMLファイル」を参照してください。

4.4.2. 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ファイル) を配置するHTTPサーバのURLに変更します。(ADFについては、「4.8. 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 ファイル、ADF、ダウンロード用 HTML ファイルを作り直します。
もし、ダウンロード用 HTML ファイル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 ファイル中の 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:¥iDKStar2.0¥apps¥<プロジェクト名>¥bin¥<プロジェクト名>.jar
 - C:¥iDKStar2.0¥apps¥<プロジェクト名>¥bin¥<プロジェクト名>.jam
 - C:¥iDKStar2.0¥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、“**ap02A-1.0**” の固定値を返します)、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。)

4.5. 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 4-15）。^[7]



図 4-15 プロジェクト読み込み画面

2. プロジェクトの格納先（C:\¥iDKStar2.0¥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。
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 4-16）。

^[7] プロジェクト一覧の「情報」欄は、通常は ADF の AppName の値が表示されます。
（ADF については、「4.8. 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、メインクラス（AppClass に設定されるクラス）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 StarApplication {
    :
```

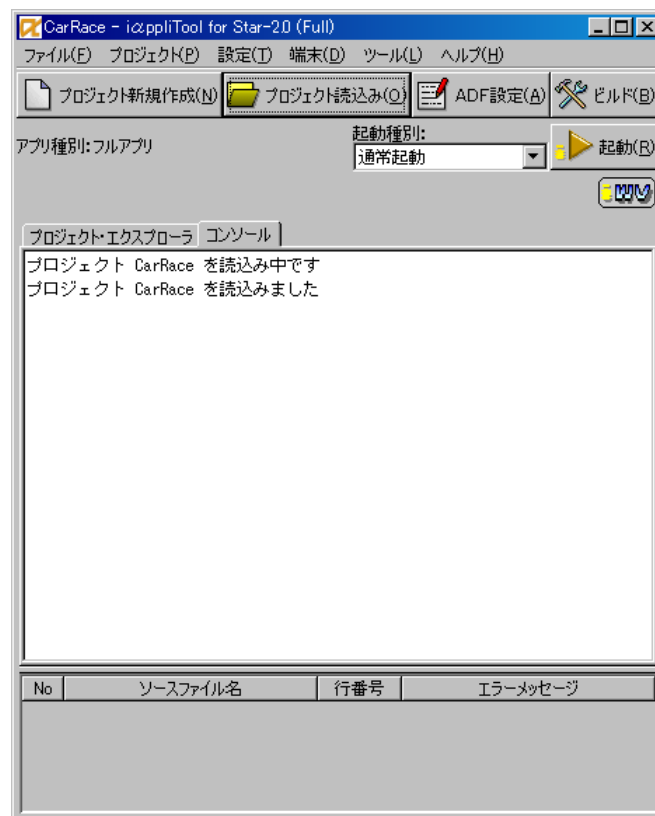



図 4-16 既存プロジェクト読み込み後のメイン画面

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。

4.5.1. 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、「4.8. 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、✖が表示されます（図 4-17）。

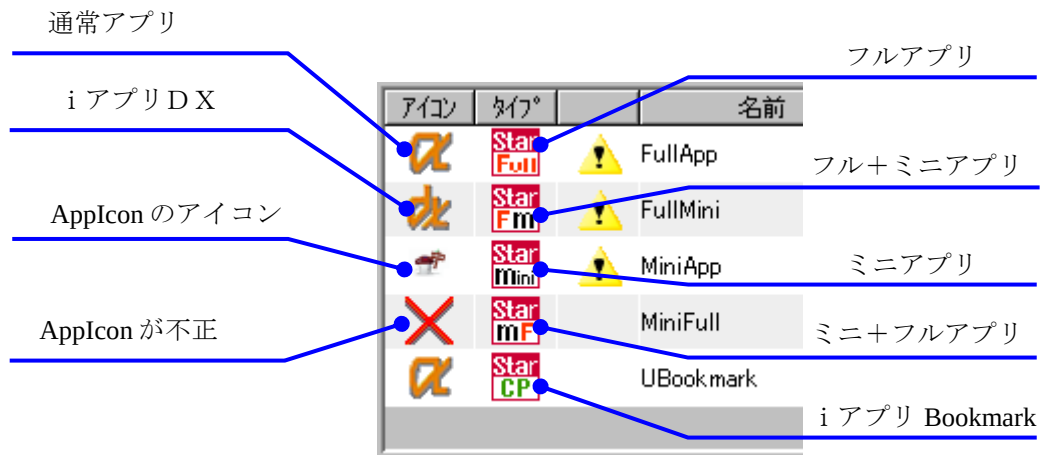


図 4-17 アイコン表示

【注意】

各プロジェクトの JAR ファイルが「iα ppli Development Kit for Star-2.0」の iα ppliTool を用いて作成されたものである場合、⚠アイコンが表示されます。⚠アイコンが表示されている場合、その JAR ファイルは Star-1.3 プロファイル以降搭載の携帯電話でのみ動作します。それ以前の携帯電話では i アプリを実行できません。

Star-1.3 プロファイル搭載機種以前の携帯電話で動作する必要がある場合は、「iα ppli Development Kit for Star-1.2」の iα ppliTool をご利用ください。

4.6. 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 4-18）。

The screenshot shows the 'プロジェクト設定' (Project Settings) window with the 'アプリケーション動作設定' (Application Action Settings) tab active. The window is divided into several sections:

- メディアリソースプロパティ設定** (Media Resource Property Settings): Includes a checkbox for '拡張HTTPヘッダの関連付け' (Link extension HTTP header) and a text field for 'X-Dcm-Move' with the value '0000'.
- アプリケーション端末移動情報設定** (Application Terminal Movement Information Settings): Includes a checkbox for 'アプリ移動情報フラグ' (App movement information flag) and a checkbox for '異なる端末間の移動' (Movement between different terminals).
- 自動起動時刻登録設定** (Automatic startup time registration settings): Includes a dropdown for 'ONETIME', fields for 'YYYY: 1999', 'Dec', 'DD: 9', 'HH: 9', and 'MM: 9'. Below these is a 'SpanSet' section with a list box showing months (January, February, March, April). A list of scheduled times is displayed: 'Onetime 1999 December 09 09:09', 'Every Monday 2011 September 12 02:02', 'Every 01 of month 2011 September 01 12:00', and 'Every July 2011 July 01 15:30'.
- 招集メッセージ起動設定** (Recruitment message startup settings): Includes a text field for '招集メッセージパラメータ' (Recruitment message parameter) and a dropdown for '招集メッセージ通信結果' (Recruitment message communication result) set to 'true'.
- 画面表示設定** (Screen display settings): Includes a text field for 'フォントベースライン設定' (Font baseline setting).

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

図 4-18 プロジェクトの設定（アプリケーション動作設定）

2. 必要な設定項目を設定します。
「アプリケーション動作設定」（図 4-18）、「ネイティブデータ設定」（図 4-19）、「デバイス設定」（図 4-20）の各タブを選択することで、それぞれの設定項目を表示・設定できます。

3. 「設定」または「適用」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、表 4-3、表 4-4 に示すとおりです。

表 4-2 プロジェクトの設定項目（アプリケーション動作設定）

設定項目	説明	デフォルト
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	StarApplicationManager#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	StarApplicationManager#isMovedFromOtherTerminal()で取得できる状態を設定します。	チェックなし (false)
自動起動時刻登録設定 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。 開始～終了の範囲は Ctrl キーを押しながら複数選択することができます。	Onetime 1999 January 09 09:09 Every Monday 02:02 Every 01 of month 12:00 Every July 01 15:30
招集メッセージ起動設定	招集メッセージパラメータと招集メッセージ通信結果を設定します。	true
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

プロジェクト設定

アプリケーション動作設定 | ネイティブデータ設定 | デバイス設定

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: 100 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

設定 取消 適用

図 4-19 プロジェクトの設定（ネイティブデータ設定）

表 4-3 プロジェクトの設定項目（ネイティブデータ設定）

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。	100
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

プロジェクト設定

アプリケーション動作設定 | ネイティブデータ設定 | **デバイス設定**

カメラデバイス情報設定

カメラデバイス数: カメラフレームレート(fps):

フォーカス制御: ☐ ハードウェア機構 ☒ ソフトウェア機構

3Dサウンド設定

3Dサウンドリソース数:

加速度センサ設定

デバイス数: 蓄積可能データ上限数:

電子コンパス設定

蓄積可能データ上限数: 計測間隔解像度: ミリ秒

歩数計情報設定

蓄積可能データ最大日数:

ジャイロセンサ設定

蓄積可能データ上限数: 計測間隔解像度: ミリ秒

キャリブレーションタイムアウト値: ミリ秒

音声合成設定

最大出力バイト数: 最大データ数: 文字の読み上げ速度: ミリ秒

設定 取消 適用

図 4-20 プロジェクトの設定 (デバイス設定)

表 4-4 プロジェクトの設定項目 (デバイス設定)

設定項目	説明	デフォルト
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
フォーカス制御	カメラフォーカスの切り替えがハードウェアによる機構であるかソフトウェアによる機構であるかを設定します。	ソフトウェア機構
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
(加速度センサ設定) デバイス数	加速度センサ有効デバイス数を 1 ~ 2 の範囲で設定します。	2
(加速度センサ設定) 蓄積可能データ上限数	加速度センサデータの最大保存件数を設定します。	20
(電子コンパス設定) 蓄積可能データ上限数	電子コンパスデータの最大保存件数を設定します。	3000

設定項目	説明	デフォルト
(電子コンパス設定) 計測間隔解像度	電子コンパスの計測間隔を設定します。	100
(歩数計設定) 蓄積可能データ最大日数	歩数計データの最大保存日数を設定します。	30
(ジャイロセンサ設定) 蓄積可能データ上限数	ジャイロセンサデータの最大保存件数を設定します。	3000
(ジャイロセンサ設定) 計測間隔解像度	ジャイロセンサの計測間隔を設定します。	100
(ジャイロセンサ設定) キャリブレーションタイムアウト値	ジャイロセンサのキャリブレーションタイムアウト値を設定します。	10000
(音声合成設定) 最大出力バイト数	音声合成機能で音声出力する文字列データに設定できる最大バイト数を設定します。	128
(音声合成設定) 最大データ数	音声合成機能でルビ情報として登録できる最大のデータ数を設定します。	16
(音声合成設定) 文字の読み上げ速度	音声合成機能の文字の読み上げ速度を設定します。	200

詳細については、「15.28. 「アプリケーション名.properties」 ファイル」を参照してください。

4.7. 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 4-21）。

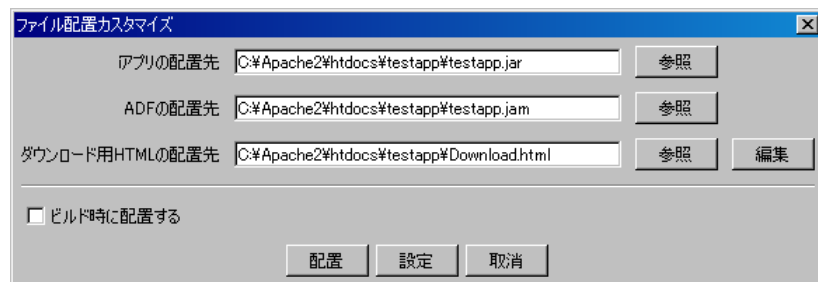


図 4-21 ファイル配置カスタマイズ画面

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 4-5 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。
実際の携帯電話からアクセス可能な Web アプリケーションサーバ機能を持つマシンで「i apli Development Kit for Star-2.0」を使用する場合、ここに公開用フォルダを指定することで、「i apli Development Kit for Star-2.0」で作成された i アプリが、そのまま公開用の環境にも反映されている状態となります。

表 4-5 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

4.8. 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。

本設定は、DoJa 向けの i アプリでは利用できません。

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 4-22)。

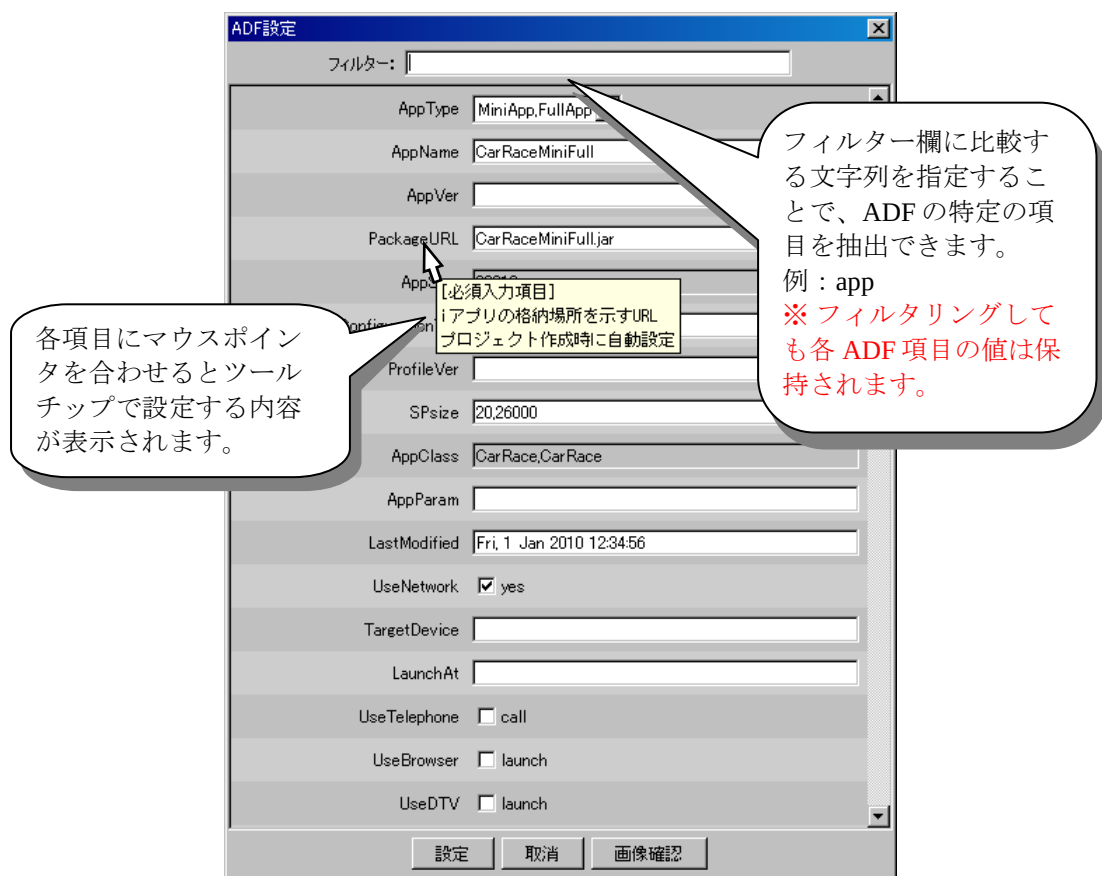


図 4-22 ADF 設定画面

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。
以降に各設定項目の内容を説明します。

AppType

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリのタイプ（フルのみアプリ、ミニのみアプリ、フル+ミニアプリ）を表す文字列。 設定する文字列は、ASCII 形式で、“ FullApp ”（フルアプリ）、“ MiniApp ”（ミニアプリ）または、“ FullApp,MiniApp ”（フル+ミニアプリ：フルアプリの起動を優先）、“ MiniApp,FullApp ”（ミニ+フルアプリ：ミニアプリの起動を優先）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。	必須

AppName

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリの表示名称（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、 StarApplication が実装されたクラス名が設定されます。 （ただし、 StarApplication 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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリ、JAR パッケージライブラ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、「6.1. 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、 <u>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。</u>	必須

InnerPackageName

対象アプリ	説明	種別
ミニ+フル フル+ミニ	JAR パッケージ(本体の JAR ファイル)に含まれる実行ファイル(JAR ファイル)のファイル名（最大 255 バイト）。 設定する文字列は、ASCII 形式とし、 i アプリの場合は、AppType に指定した順番に従って実行ファイルを 2 つ指定します。 もし、指定した数が AppType と異なる場合は、アプリケーションの起動を禁止します。 【例】 AppType = FullApp, MiniApp InnerPackageName = Full.jar, Mini jar InnerAppSize = 10240, 5120 （フルアプリ用の実行ファイル名は Full.jar で 10240 バイト、ミニアプリ用の実行ファイル名は Mini.jar で 5120 バイトとなります。）	オプション

InnerAppSize

対象アプリ	説明	種別
ミニ+フル フル+ミニ	1 つの JAR ファイル内に複数の実行ファイルを持つ場合における、各実行ファイルのサイズ（バイト数）。 InnerPackageName に指定した順番に従って JAR ファイルのサイズを 2 つ指定します。もし、ファイルサイズの数 が AppType と異なる場合は、アプリケーションの起動を禁止します。 【例】 AppType = FullApp, MiniApp InnerPackageName = Full.jar, Mini jar InnerAppSize = 10240, 5120 （フルアプリ用の実行ファイル名は Full.jar で 10240 バイト、ミニアプリ用の実行ファイル名は Mini.jar で 5120 バイトとなります。）	オプション

AppSize

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリのサイズ（バイト単位）。 【サイズ制限】 ●ミニアプリのみ：50KByte まで。 ●フルアプリ、ミニ+フル、フル+ミニ：Ssize との合計が 10,240KByte まで。 実際の i アプリ（JAR ファイル）のサイズと違っていたり、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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	コンフィグレーションのバージョン。 設定する文字列は、“CLDC-1.1” 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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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。複数のプロファイルバージョンを指定する場合は、カンマ(,)で区切って指定します。 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。 ・“Star-1.0”（Star-1.0 シリーズ対応） ・“Star-1.1”（Star-1.1 シリーズ対応） ・“Star-1.2”（Star-1.2 シリーズ対応） ・“Star-1.3”（Star-1.3 シリーズ対応） ・“Star-1.5”（Star-1.5 シリーズ対応） ・“Star-2.0”（Star-2.0 シリーズ対応）	オプション

SPsize

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	スクラッチパッドのサイズ（バイト単位）。 【サイズ制限】 ●ミニアプリのみ：200KByte まで。 ●フルアプリ、ミニ+フル、フル+ミニ ： AppSize との合計が 10,240KByte まで。 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。（フルアプリ、ミニ+フル、フル+ミニアプリは最大 16 個、ミニアプリは 1 個のみ。ミニ+フル、フル+ミニアプリにおいて、ミニアプリは、最初の 1 番目のみ使う事ができます。） （例：16, 1024） 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。 ミニ+フル、フル+ミニアプリでは、ミニアプリからは最初のスクラッチパッドのみアクセス可能です。 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 4-22）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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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、<code>com.docomostar.StarApplication</code> 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。 AppType に指定した順番に従って、フルアプリとミニアプリのメインクラスを指定します。もし、ファイルサイズの数が AppType と異なる場合は、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、<code>StarApplication</code> が実装されたクラス名が設定されます。（ただし、<code>StarApplication</code> 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。） ミニ+フルアプリ、フル+ミニアプリにおいては、プロジェクトのビルド時にパッケージされているフルアプリ、ミニアプリの AppClass の値を自動的に設定します。そのため、画面からこの項目を変更する事はできません。	必須

AppParam

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリの起動時に渡すパラメータ（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、<code>com.docomostar.StarApplicationManager.getArgs()</code> 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 <code>null</code> 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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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「4.8.3. 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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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 "yes"（ASCII 形式とし、大文字・小文字の区別はしません）のみとし、それ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場合例外が発生します。 <u>この値は、DoJa の時と異なりますので注意が必要です。</u>	オプション

TargetDevice

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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）。 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、 ” X01A, Y01A, Z01A ” 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にしているとみなされます。 なお、2008 年秋以降に発売される NTT ドコモの携帯電話ではそれまでのシリーズの概念がなくなり、型番の付与方針が変更されています。これに伴い、それ以前の機種でサポートされていたシリーズ全機種の一括指定（all 指定）は、2008 年秋以降発売の機種では廃止されていますので注意してください。 ただし、一括指定（all 指定）が廃止されたことに伴い、それ以前の機種で設けられていた本項目の 128 バイトという上限も撤廃され、より多数の機種を列挙できるようになりました。 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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	起動時間間隔指定。（有効桁数は 2 桁まで） 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

UseTelephone

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	メーラから 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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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）。 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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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。 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 “bt:”を設定した場合は Bluetooth 機器 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、先頭の “Irda:”、“code:”、“icc:”を含め 255 バイト以内である必要があります)。 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、各外部インタフェース毎に先頭の 1 つのみ有効とします。	オプション

AppTrace

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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] $96 \leq [\text{横のピクセル数}] \leq \text{携帯電話の最大横ピクセル数}$ $72 \leq [\text{縦のピクセル数}] \leq \text{携帯電話の最大縦ピクセル数}$ i アプリ起動時の画面を、携帯電話の描画領域の中央に設定します。（x は半角小文字のエックス (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、通常どおりの描画とします。	オプション

Pallet

対象アプリ	説明	種別
ミニ ミニ+フル フル+ミニ	ミニアプリが使用するパレットフォーマット。 以下の形式で、コロン(:)で区切って値を指定します。 PalletSize : PalletDrawArea : PositionX : PositionY : PalletFace • PalletSize パレットのサイズを” (横のドット数) x (縦のドット数) ” の形で指定。省略時は 160x160。 • PalletDrawArea パレット上の描画領域のサイズを” (横のドット数) x (縦のドット数) ” の形で指定。省略時は PalletSize。 • PositionX、PositionY パレット上の描画領域を表示する位置(原点)を指定。省略された場合はそれぞれ 0、0。 • PalletFace パレットのフェイスとなる画像を指定。省略された場合はフェイス指定なし。 PalletDrawArea に指定可能な最小値は以下のとおりです。 横 : 96 縦 : 40 PalletSize に指定可能な値は、以下のとおりです。 440x80、320x240、240x320、160x160 PalletDrawArea、PositionX、PositionY に指定可能な値は、以下の条件を満たしていなければなりません。 • 横方向 : PalletDrawArea の横のドット数 + PositionX ≤ PalletSize の横のドット数 • 縦方向 : PalletDrawArea の縦のドット数 + PositionY ≤ PalletSize の縦のドット数 上記の形式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。	オプション

GetSysInfo

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	NTT ドコモで付与するトラステッド i アプリの管理 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 (i アプリ DX) 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 "000000000000" 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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	トラステッドの最終更新日時	オプション

PackageHashCode

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	JAR ファイルの Hash 値。 PackageURL で指定された JAR ファイルの 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、「9.8. CPBindingAuth署名付与ツール」を参照してください。	オプション

LaunchByApp

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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例が発生します。	オプション

MessageCode

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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、トルカデータ登録・参照、歩数計の歩数情報の参照、身体活動量計の情報の参照、身体情報の参照、フォーム分析情報の参照、ミュージック情報管理機能の使用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。 この項目に何も設定しない場合は、アプリケーションから、上記の機能を実行しようとする例が発生します。	オプション

GetUtn

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	携帯電話及び 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" 携帯電話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

LaunchApp

対象アプリ	説明	種別
フル フル+ミニ ミニ+フル	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

FeliCaCPID

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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 4-23 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。 ※ 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 を関連付けることができます (「4.8.2 FeliCa フリー領域の関連付け」を参照)。	フリー領域用 API を使用する場合は必須

FeliCaPIN

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	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。 ※ 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（「4.8.2 FeliCaフリー領域の関連付け」を参照）。	フリー領域用 API を使用する場合は必須

AppIcon

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	アプリケーションの一覧でアイコンやサムネイルとして使用する画像ファイル名。（最大 255 バイト） アイコンとして使用可能な画像は、1 番目の指定が 48×48、2 番目の指定が 96×96 ドットの GIF、PNG または JPEG 画像です。（2 番目の指定が無い場合は、1 番目の画像を 2 倍に拡大表示します。） 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。ただし 3 番目以降は無視します。 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。 ADF 設定画面の「画像確認」ボタンを押すことで、AppIcon キーに設定されている画像の確認が行えます（「4.8.4 アイコン・タイトル画像、起動時画像の確認方法」を参照）。	オプション

UseStorage

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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、「9.8. CPBindingAuth署名付与ツール」を参照してください。	オプション

CPName

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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 SJIS 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、各携帯電話に依存します。	オプション

GetLocationParam

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	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

AppMainTitle

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	アプリ一覧でグラフィカル表示をする場合のメインタイトル画像ファイル名。（最大 255 バイト） メインタイトル画像として使用可能な画像は、160×160 ドットまたは 320×320 ドットの GIF、PNG または JPEG 画像です。 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。1 番目には 160×160 ドット、2 番目には 320×320 ドットの画像を指定します。3 番目以降は無視します。 2 件以上の画像が指定された場合は 2 番目の画像（320×320 ドットの画像）が使用されます。1 件のみ指定された場合は、1 番目の画像を 320×320 ドットに拡大して使用されます。 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。 ADF設定画面の「画像確認」ボタンを押すことで、AppMainTitleキーに設定されている画像の確認が行えます（「4.8.4 アイコン・タイトル画像、起動時画像の確認方法」を参照）。	オプション

DeniedMultiApp

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリとの同時起動を許可しないネイティブアプリ名。（最大 255 バイト） 音楽プレイヤー（ミュージックチャネルプレイヤーも含む）との同時起動を許可しない場合は、 "music" ワンセグアプリとの同時起動を許可しない場合は、 "dtv" 音楽プレイヤー、及びワンセグアプリのいずれとも同時起動を許可しない場合は、 "all" を指定します（ASCII 形式とし、大文字・小文字の区別はしません）。 複数のネイティブアプリを指定する場合は、“,” で区切って設定します。	オプション

UseMSO

対象アプリ	説明	種別
フル フル+ミニ ミニ+フル	アプリケーションが SWF ファイルを再生する場合のデータストレージ領域(MSO)として使用するスクラッチパッドの番数。 指定された整数値が N の場合、SPsize キーで指定された第 N 番目のスクラッチパッドをデータストレージ領域 (MSO) として扱います。	オプション

MenuTitleImage

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	未設定状態の IC アプリのアイコン・サムネイル画像、タイトル画像として使用する画像ファイル名。（最大 255 バイト） 使用可能な画像は、1 番目の指定が 96×96 ドット（アイコン・サムネイル画像用）、2 番目の指定が 160×160 ドット（タイトル画像用）、3 番目の指定が 320×320 ドット（タイトル画像用）の GIF、PNG または JPEG 画像です。 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。ただし 4 番目以降は無視します。 3 件以上の画像が指定された場合、タイトル画像としては 3 番目の画像（320×320 ドットの画像）が使用されます。 2 件のみ指定された場合は、2 番目の画像を 320×320 ドットに拡大して使用されます。 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

AppLabel

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリが利用する特定デバイスや機能。 歩数計を利用する場合は、 “pedometer” 身体活動量計を利用する場合は、 “physicalactivitymonitor” 身体情報を利用する場合は、 “bodyprofiles” フォーム分析情報を利用する場合は、 “formalyzer” を指定します（ASCII 形式とし、大文字・小文字の区別はしません）。 複数の特定デバイス、機能を指定する場合には、空白、タブで区切って設定します。	オプション

LaunchBySchedule

対象アプリ	説明	種別
フル フル+ミニ ミニ+フル	スケジューラからの i アプリ起動許可。 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。	オプション

LaunchByDataBox

対象アプリ	説明	種別
フル フル+ミニ ミニ+フル	データ BOX からの i アプリ起動が可能なファイル種別。 画像からの i アプリ起動を利用する場合は、 "picture" を指定します (ASCII 形式)。	オプション

RestrictedAudioOutput

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	外部音声出力機器への i アプリからの音の出力可否。 Bluetooth ヘッドフォン等の A2DP 対応オーディオシンクへ i アプリからの音を出力させない場合は、 "bluetooth" を指定します (ASCII 形式とし、大文字・小文字の区別はしません)。	オプション

VmOption

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	Java VM のオプション設定。 i アプリから System.gc() を実行したタイミングでガベージコレクションを実施したい場合は、 "gc: on" を指定します (ASCII 形式とし、大文字・小文字の区別はしません)。指定がない場合は i アプリから実行した System.gc() は無視され、Java VM が必要と判断したタイミングでガベージコレクションを実施します。なお、 "gc: on" を指定した場合、System.gc() 実行時の性能が低下する可能性があります。	オプション

LaunchByFaceApp

対象アプリ	説明	種別
フル ミニ+フル フル+ミニ	MyFACE 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

StartingAppImage

対象アプリ	説明	種別
フル ミニ ミニ+フル フル+ミニ	i アプリ 起動時に表示する画像ファイル名。（最大 255 バイト） 起動時画像として使用可能な画像は、JPEG、BMP、GIF（アニメーション GIF を含む）、PNG 画像です。 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。1 番目に縦画面用、2 番目に横画面用の起動時画像ファイルを指定します。3 番目以降は無視します。 ファイル名を指定せずにカンマのみ入力した場合、起動時画像が無指定とみなします。1 番目を指定しなかった場合（指定文字列の先頭が “,” の場合）は縦画面時に起動時画像が表示されなくなります。2 番目を指定しなかった場合（1 番目のみ指定した場合など）は横画面時に起動時画像が表示されなくなります。1 番目、2 番目の両方を省略することも可能です。 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。 ADF 設定画面の「画像確認」ボタンを押すことで、StartingAppImage キーに設定されている画像の確認が行えます（「4.8.4 アイコン・タイトル画像、起動時画像の確認方法」を参照）。	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:¥iDKStar2.0¥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。

4.8.1. FeliCaCPID の設定

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 4-23 で説明します。

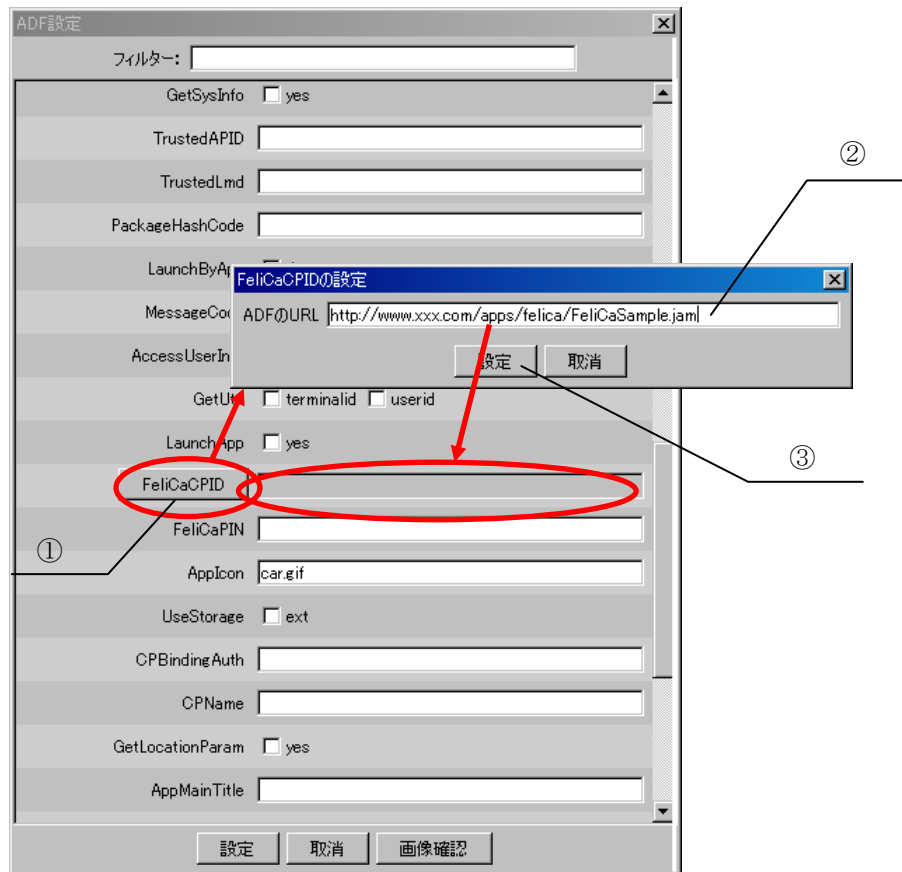


図 4-23 FeliCaCPID 設定画面

- 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 4-23）。
- ② FeliCaCPID 設定画面の「ADF の URL」に ADF 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「アプリケーション名.properties」ファイルに保持され次回から「ADFのURL」に表示されます（「15.28 「アプリケーション名.properties」ファイル」参照）。
- 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。

4.8.2. 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「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 4-24）。

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 4-24 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。

4.8.3. 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。

4.8.4. アイコン・タイトル画像、起動時画像の確認方法

ADF に AppIcon、AppMainTitle、StartingAppImage が設定されている場合、各設定項目に指定された画像ファイルをチェックすることができます。

画像ファイルをチェックするには：

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 4-25）。

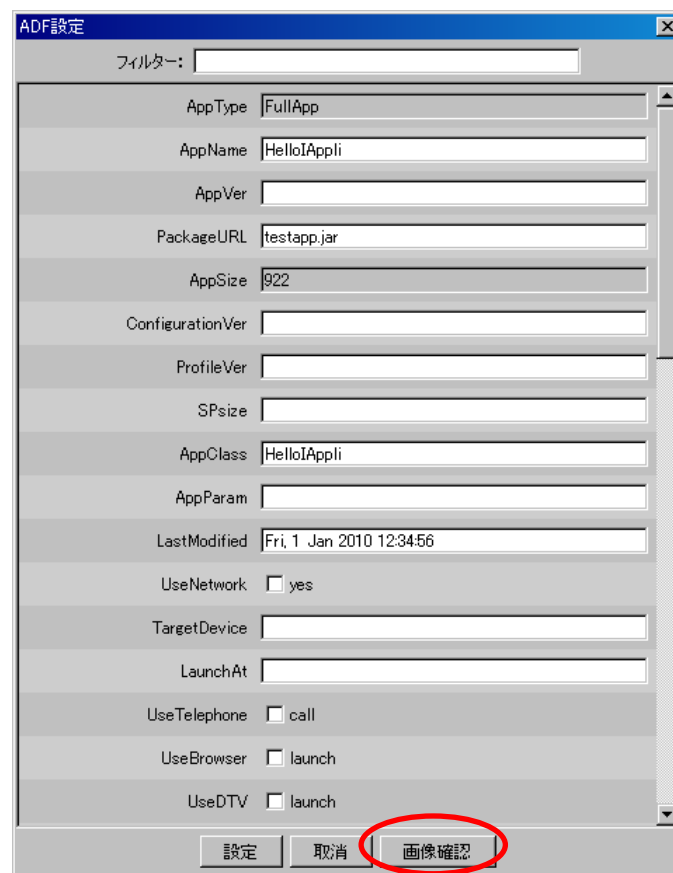


図 4-25 ADF 設定画面

2. 「ADF設定」画面で「画像確認」ボタンを押します。
「画像確認」画面が表示され、現在ADFのAppIcon、AppMainTitleに設定されている画像イメージが表示されます（図 4-26）。



図 4-26 画像確認画面

- 「アイコン画像」には、ADF の AppIcon に指定したアイコンが表示されます。AppIcon のアイコンは、48×48 ドットまたは 96×96 ドットのサイズで作成された GIF、PNG または JPEG 形式のファイルです（96×96 ドット画像を表示します。48×48 ドット画像のみ指定されている場合は 96×96 ドットに拡大表示します）。
 - 「メインタイトル画像」には、ADF の AppMainTitle に指定したタイトル画像が表示されます。AppMainTitle の画像は、160×160 ドットまたは 320×320 ドットのサイズで作成された GIF、PNG または JPEG 形式のファイルです（320×320 ドット画像を表示します。160×160 ドット画像のみ指定されている場合は 320×320 ドットに拡大表示します）。
 - それぞれの画像ファイルがプロジェクトの res 配下におかれ、アプリケーションのビルドによって JAR ファイルの中に含まれている必要があります。
3. 「起動時画像（縦）確認」ボタンを押すと、「画像確認（起動時画像：縦）」画面が表示され、現在の ADF の StartingAppImage の 1 番目に指定されている画像イメージが表示されます。
また「起動時画像（横）確認」ボタンを押すと、「画像確認（起動時画像：横）」画面が表示され、現在の ADF の StartingAppImage の 2 番目に指定されている画像イメージが表示されます。

- **StartingAppImage** の画像は、JPEG、BMP、GIF（アニメーション GIF 含む）、PNG のいずれかの形式です。画像サイズに制限はありません。

AppIcon と **AppMainTitle** に指定した画像は、携帯電話の i アプリ一覧画面をグラフィカル表示状態にしたときに表示されます。また **StartingAppImage** に指定した画像は、携帯電話で i アプリを起動するときに表示されます。画像確認画面で期待どおりの画像が表示されない場合、携帯電話へのダウンロードに失敗するか、i アプリ一覧画面で画像が正しく表示されない可能性がありますので、上記の表示条件をもう一度確認してください。

4.9. トラストッド動作の設定

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

トラストッド 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
- Open GL/ES を使った機能
- 音声認識を使った機能
- FeliCa を使った機能（一部を除く）
- TCP/UDP を使った機能
- 招集メッセージによる起動
- 地図プラットフォームの起動
- クラスローダを使った機能
- マイメニューの登録/削除
- 個別課金コンテンツに関する機能
- i アプリダウンロード元以外のサーバのアプリケーション連携機能
- RAW 画像を使った機能
- オート GPS に関する機能
- ケータイ払い／ポイント払いに関する機能

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、トラストッド 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。

トラストッド動作を設定するには：

1. メニューから「設定」→「トラストッド動作設定...」を選択します。
「トラストッド動作設定画面」が表示されます（図 4-27）。

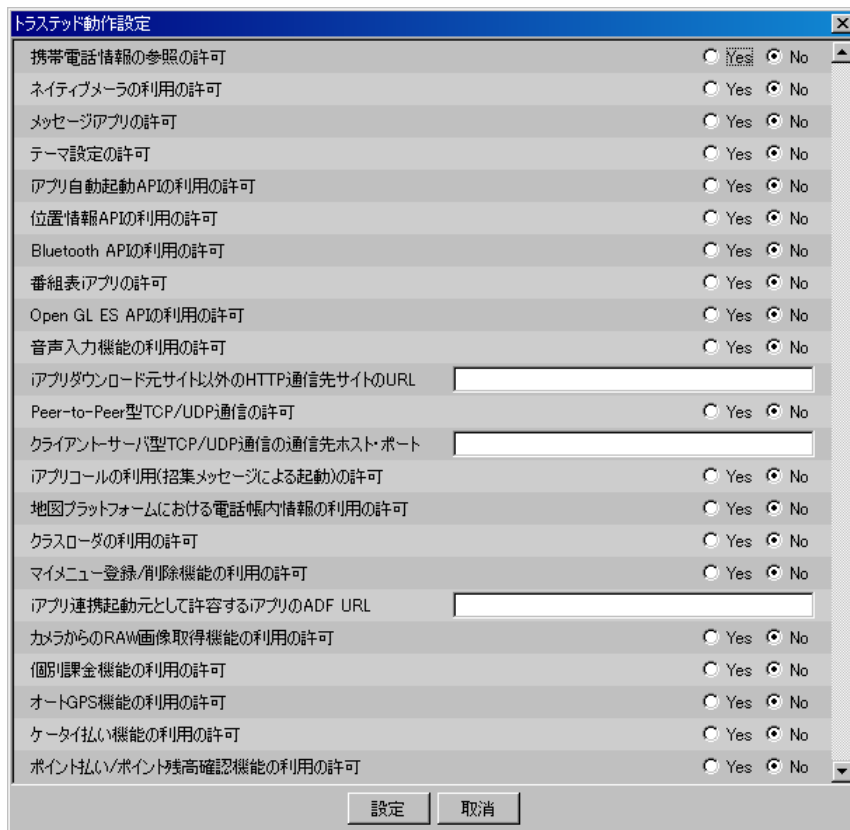


図 4-27 トラステッド動作設定画面

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 4-6 に示すとおりです。

表 4-6 トラステッド動作設定の設定項目

設定項目	説明
携帯電話情報の参照の許可	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 API を利用する場合に Yes に変更します。 (関連：PhoneBook、PhoneBookGroup、CallRecord、MailAgent、Schedule、XString)
ネイティブメーラの利用の許可	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：MailAgent、MailAgent2、XString)
メッセージ i アプリの許可	メッセージ i アプリの API を利用する場合に Yes に変更します。 (関連：MessageAgent、XString)
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 API を利用する場合に Yes に変更します。 (関連：PhoneTheme、SoundStore、AvatarStore、ImageStore)

設定項目	説明
i アプリ自動起動 API の利用の許可	アプリケーションの自動起動をする API を利用する場合に Yes に変更します。 (関連：StarApplicationManager(setLaunchTime、getLaunchTime))
位置情報 API の利用の許可	位置情報を扱う API を利用する場合に Yes に変更します。 (関連：LocationProvider、CityInfoStore、LocationRecord、PhoneBook、「4.19 位置情報起動」、「4.20 位置情報画像起動」、「4.21 位置情報電話帳起動」、「4.29 コピー文字列起動」)
Bluetooth API の利用の許可	Bluetooth の API を利用する場合に Yes に変更します。 (関連：Bluetooth)
番組表 i アプリの許可	DTV 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。 (関連：「4.26 DTV起動」)
Open GL ES API の利用の許可	OpenGL/ES の API を利用する場合に Yes に変更します。(関連：com.docomostar.ui.ogl.*)
音声入力機能の利用の許可	音声認識機能の API を利用する場合に Yes に変更します。 (関連：SpeechRecognizer、DDPConnection)
i アプリダウンロード元サイト以外の HTTP 通信先サイトの URL	ダウンロード元以外に i アプリが通信を許可する URL 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。 複数の URL を指定するには、「□」(空白)で区切って指定してください。
Peer-to-Peer 型 TCP/UDP 通信の許可	TCP/UDP 機能で、i モード携帯電話同士の通信をおこなう場合に Yes に変更します。 (関連：SocketConnection, UDPDatagramConnection)
クライアント-サーバ型 TCP/UDP 通信の通信先ホスト・ポート	TCP/UDP 機能で、通信を許可する URL を“ホスト名：ポート番号”で設定します。 複数の URL を指定するには、「□」(空白)で区切って指定してください。
i アプリコールの利用(招集メッセージによる起動)の許可	招集メッセージにより、i アプリを起動する場合に Yes に変更します。 (関連：「4.28 招集メッセージ起動」)
地図プラットフォームにおける電話帳内情報の利用の許可	地図プラットフォームとして連携起動する事を許可する場合に Yes に変更します。 (関連：「4.27 地図プラットフォーム起動/ナビ起動」)
クラスローダの利用の許可	ByteArrayClassLoader の API を利用する場合に Yes に変更します。 (関連：ByteArrayClassLoader)
マイメニュー登録/削除機能の利用の許可	マイメニュー登録をする事を許可する場合に Yes に変更します。 (関連：Contents)

設定項目	説明
i アプリ連携起動元として許容する i アプリの ADF URL	i アプリダウンロード元以外のサーバのアプリケーションを連携起動する場合に Yes に変更します。
カメラからの RAW 画像取得機能の利用の許可	RAW 画像を扱う API を利用する場合に Yes に変更します。 (関連：RawImageCapture) ※RAW画像取得機能を使用する際には、事前に画像サイズを設定し、それに合わせたJPEG画像データを用意する必要があります。詳細は「15.52 RAW画像取得機能関連ファイル」を参照ください。
個別課金機能の利用の許可	個別課金機能の使用を許可する場合に Yes に変更します。 (関連：Contents)
オート GPS 機能の利用の許可	オート GPS 機能の使用を許可する場合に Yes に変更します。 (関連：AutoPositioning)
ケータイ払い機能の利用の許可	ケータイ払い機能の使用を許可する場合に Yes に変更します。 (関連：Contents)
ポイント払い/ポイント残高確認機能の利用の許可	ポイント払い機能、ポイント残高確認機能の使用を許可する場合に Yes に変更します。 (関連：Contents)

ワンポイント

API リファレンスにて「トラステッド i アプリ」と指定のある API、もしくは、「SecurityException - パーミッションとして○○○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 API 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。

API リファレンスは、NTT ドコモから適宜入手してください。

4.10. 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：

- トラステッドi アプリです（「4.9. トラステッド動作の設定」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、「トラステッド動作設定」→「メッセージi アプリの許可」を Yes にした場合のみ起動可能となります（「4.8. 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）。

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（「4.4. プロジェクトのビルドと通常起動」参照）。

4.10.1. 受信フォルダ起動手順

「i alpha Tool」でi アプリを受信フォルダ起動するには：

1. メイン画面の「起動種別」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。

4.11. 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（「4.9. 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 アプリの許可」を Yes にした場合のみ起動可能となります（「4.8. 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）。

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（「4.4. プロジェクトのビルドと通常起動」参照）。

4.11.1. 送信フォルダ起動手順

「i alpha Tool」でi アプリを送信フォルダ起動するには：

1. メイン画面の「起動種別」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。

4.12. 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方法で「通常起動」とは以下の点が異なります：

- トラストッドi アプリです（「4.9. トラストッド動作の設定」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、「トラストッド動作設定」→「メッセージi アプリの許可」を Yes にした場合のみ起動可能となります（「4.8. 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）。

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（「4.4. プロジェクトのビルドと通常起動」参照）。

4.12.1. 未送信フォルダ起動手順

「i α ppliTool」でi アプリを未送信フォルダ起動するには：

1. メイン画面の「起動種別」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。

4.13. 初回起動

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

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

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（「4.4. プロジェクトのビルドと通常起動」参照）。

4.13.1. 初回起動手順

「i α ppliTool」でi アプリを初回起動するには：

1. メイン画面の「起動種別」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。

4.14. 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（「4.8. 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）。

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（「4.4. プロジェクトのビルドと通常起動」参照）。

4.14.1. タイマ起動手順

「iαppliTool」でi アプリをタイマ起動するには：

1. メイン画面の「起動種別」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。

4.15. 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のトリガで起動する方法で「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（「4.8. 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）。

4.15.1. 外部機器起動手順

「iα ppliTool」でi アプリを外部機器起動するには:

1. メイン画面の「起動種別」プルダウンから「外部機器起動」を選択し、「起動」ボタンを押します。
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 4-28）。

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。

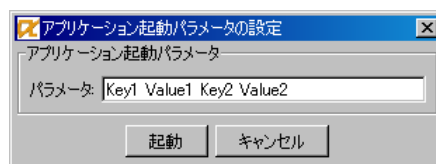


図 4-28 起動パラメータの入力画面

3. 「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

「iα ppliTool」でi アプリをメーラから起動するには：

1. 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（「15.1. メーラ起動ファイル」を参照）。
2. メイン画面の「起動種別」プルダウンから「メーラ起動」を選択し、「起動」ボタンを押します。

メーラ画面が表示されます（図 4-29）。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（「4.17. ブラウザ起動」、「4.34 MyFACE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（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. 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。

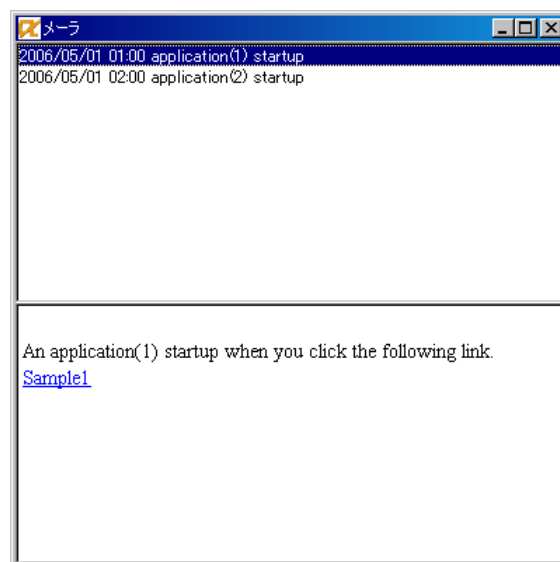


図 4-29 メーラ画面

3. 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（「4.8. 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（「15.1. 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:\¥iDKStar2.0¥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 4-30）、「はい」ボタンを押すと i アプリが起動します。

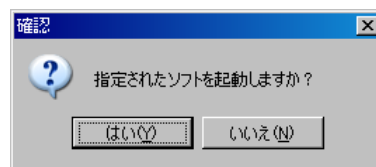


図 4-30 起動確認ダイアログ

4. i アプリを終了すると、再びメーラ画面が表示されます。
ただし、i アプリを強制終了（エミュレーション端末画面の「終話ボタン」を押して終了）した場合、メーラ画面は表示されません。

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

「iα ppliTool」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「起動種別」プルダウンから「ブラウザ起動」を選択し、「起動」ボタンを押します。

ブラウザ画面が表示されます（図 4-31）。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（「4.16 メーラ起動」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（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. DTV起動」を参照）が起動している。

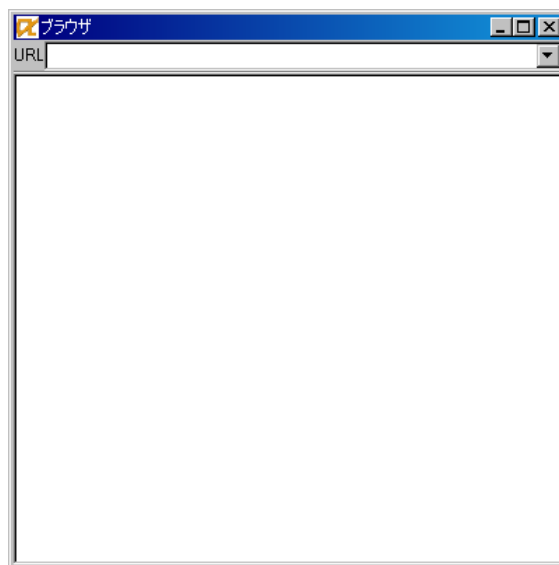


図 4-31 ブラウザ画面

2. 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（「15.4. ブラウザ起動HTMLファイル」参照）。
3. 図 4-31 の上方のURLフィールドに 2.で編集したHTMLのURLを入力し、リターンキーを押します（図 4-32）。



図 4-32 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:¥iDKStar2.0¥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:/iDKStar2.0/lib/sample.html**」と指定します。

4. 図 4-32 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（「4.8. 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（「15.4. ブラウザ起動HTML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:¥iDKStar2.0¥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 4-33）、「はい」ボタンを押すと i アプリが起動します。

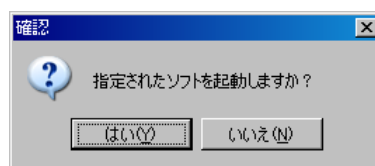


図 4-33 起動確認ダイアログ

5. i アプリを終了すると、再びブラウザ画面が表示されます。
ただし、i アプリを強制終了（エミュレーション端末画面の「終話ボタン」を押して終了）した場合、ブラウザ画面は表示されません。

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

「iα ppliTool」でi アプリをコードリーダーから起動するには：

1. 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（「15.11. コード認識結果ファイル」参照）。
2. メイン画面の「起動種別」プルダウンから「コードリーダー起動」を選択し、「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 4-34）。

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（「4.16. 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」、「4.34 MyFACE起動」を参照）が起動している。
- トルカ（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. DTV起動」を参照）が起動している。

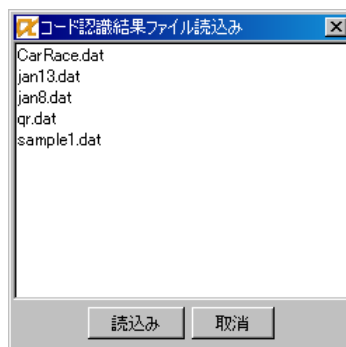


図 4-34 コード認識結果ファイル読み込み画面

3. 「図 4-34」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（「4.8. 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（「15.11. コード認識結果ファイル」）。

- 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:\¥iDKStar2.0¥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 4-35）。

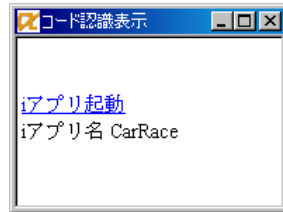


図 4-35 コード認識表示画面

4. 図 4-35 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（図 4-36）、「はい」ボタンを押すと i アプリが起動します。

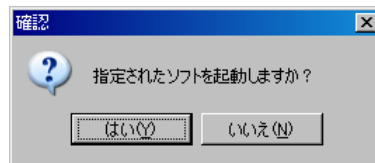


図 4-36 起動確認ダイアログ

4.18.2. CodeReader の実行手順

「iα ppli Development Kit for Star-2.0」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 4-37）。

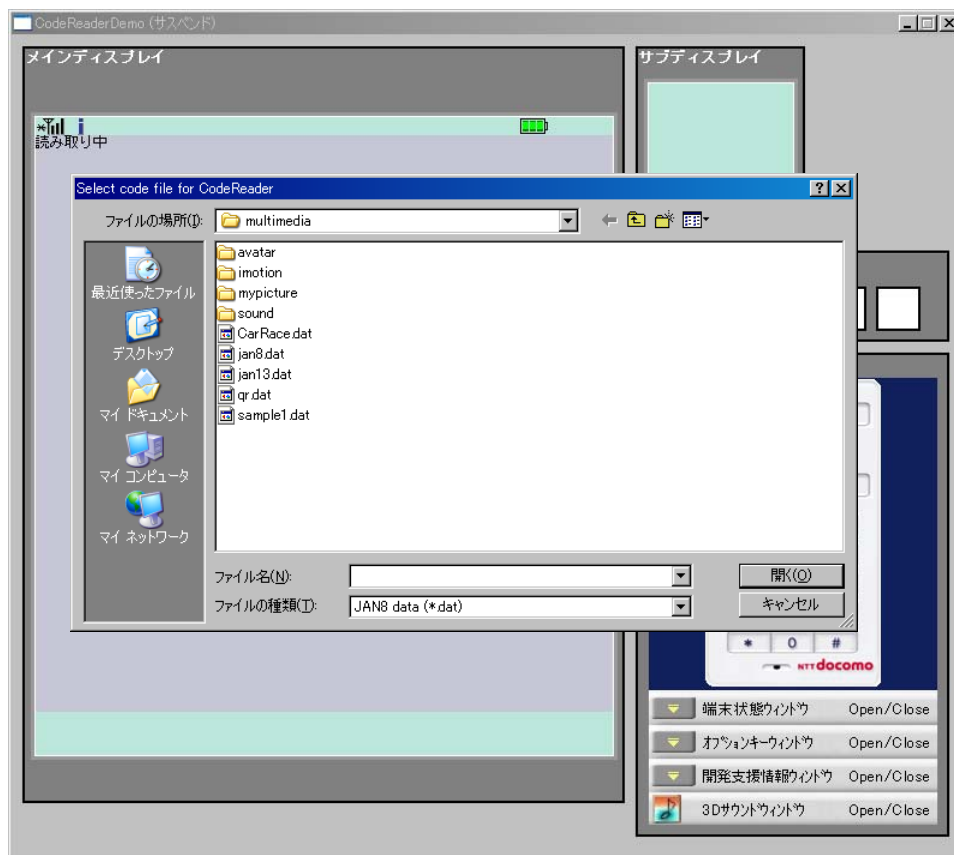


図 4-37 コード認識結果ファイル 指定ダイアログ

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：

- トラストッド i アプリです（「4.9. トラストッド動作の設定」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「TrustedAPIID」に適切な値を設定し、「トラストッド動作設定」→「位置情報APIの利用の許可」を Yes にした場合のみ起動可能となります（「4.8. 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

「i α ppliTool」で i アプリを位置情報起動するには：

1. メイン画面の「起動種別」プルダウンから「位置情報起動」を選択し、「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 4-38）。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（「4.16. 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」、「4.34 MyFACE 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（「4.23. トルカ起動」を参照）が起動している。
- BML ブラウザ（「4.25. BML 起動」を参照）が起動している。
- DTV（「4.26. DTV 起動」を参照）が起動している。

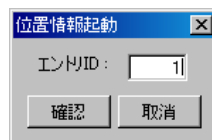


図 4-38 位置情報エントリ ID 入力画面

2. 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。
3. 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 4-39）。

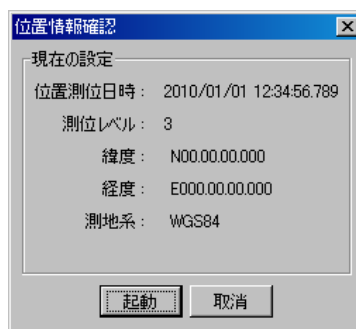


図 4-39 位置情報起動確認画面

4. 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 4-40）。

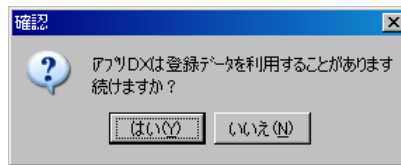


図 4-40 トラステッド起動可否

5. 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。

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 (StarApplicationManager.getParameter)	内容
“latitude”	緯度
“longitude”	経度
“datum”	測地系 (WGS84:0 Tokyo datum:1)
“level”	測位レベル
“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：

- トラストッドi アプリです（「4.9. トラストッド動作の設定」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、「トラストッド動作設定」→「位置情報APIの利用の許可」を Yes にした場合のみ起動可能となります（「4.8. 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

「i α ppliTool」でi アプリを位置情報画像起動するには：

1. メイン画面の「起動種別」プルダウンから「位置情報画像起動」を選択し、「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 4-41）。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（「4.16. 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」、「4.34 MyFACE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（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. DTV起動」を参照）が起動している。

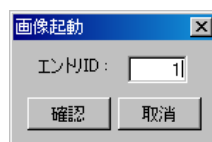


図 4-41 位置情報画像エントリ ID 入力画面

2. 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。
3. 位置情報画像確認画面が表示されるので、「起動」を押します（図 4-42）。

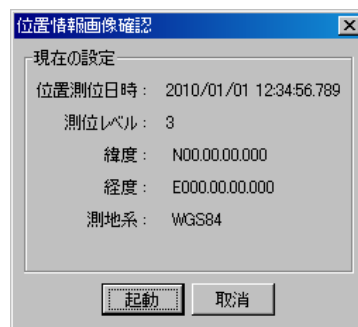


図 4-42 位置情報画像確認画面

4. 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 4-43）。

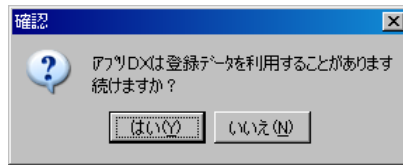


図 4-43 トラストッド起動可否

5. 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。

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：

- トラストッドi アプリです（「4.9. トラストッド動作の設定」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、「トラストッド動作設定」→「位置情報APIの利用の許可」を Yes にした場合のみ起動可能となります（「4.8. 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

「i α ppliTool」でi アプリを位置情報電話帳起動するには：

1. メイン画面の「起動種別」プルダウンから「位置情報電話帳起動」を選択し、「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 4-44）。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（「4.16. 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」、「4.34 MyFACE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（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. DTV起動」を参照）が起動している。

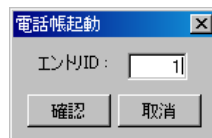


図 4-44 電話帳エントリ ID 入力画面

2. 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。
3. 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 4-45）。

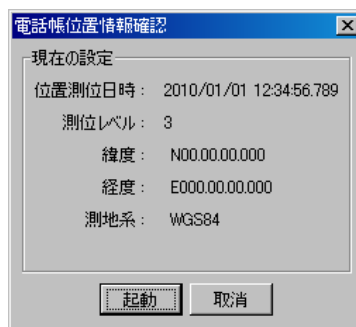


図 4-45 位置情報電話帳起動確認画面

4. 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 4-46）。

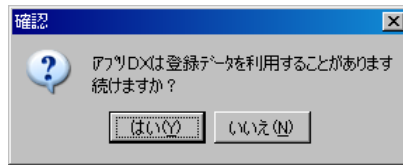


図 4-46 トラストッド起動可否

5. 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。

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、「6.7. 仮想 FeliCa 設定」を参照してください。

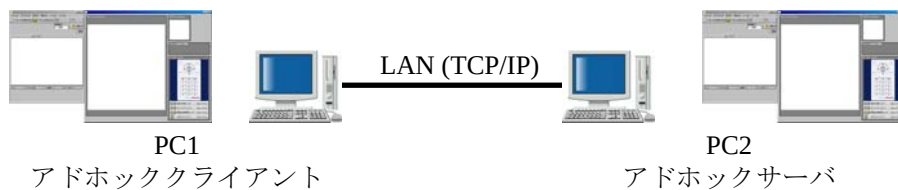


図 4-47 仮想 FeliCa アプリのアドホック通信シミュレーション

4.22.1. アドホック起動手順

「iα ppliTool」でiアプリをアドホック起動するには：

1. メイン画面の「起動種別」プルダウンから「アドホック起動」を選択し、「起動」ボタンを押します。
2. アドホック連携待機中になり、「6.7. 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 4-48）。

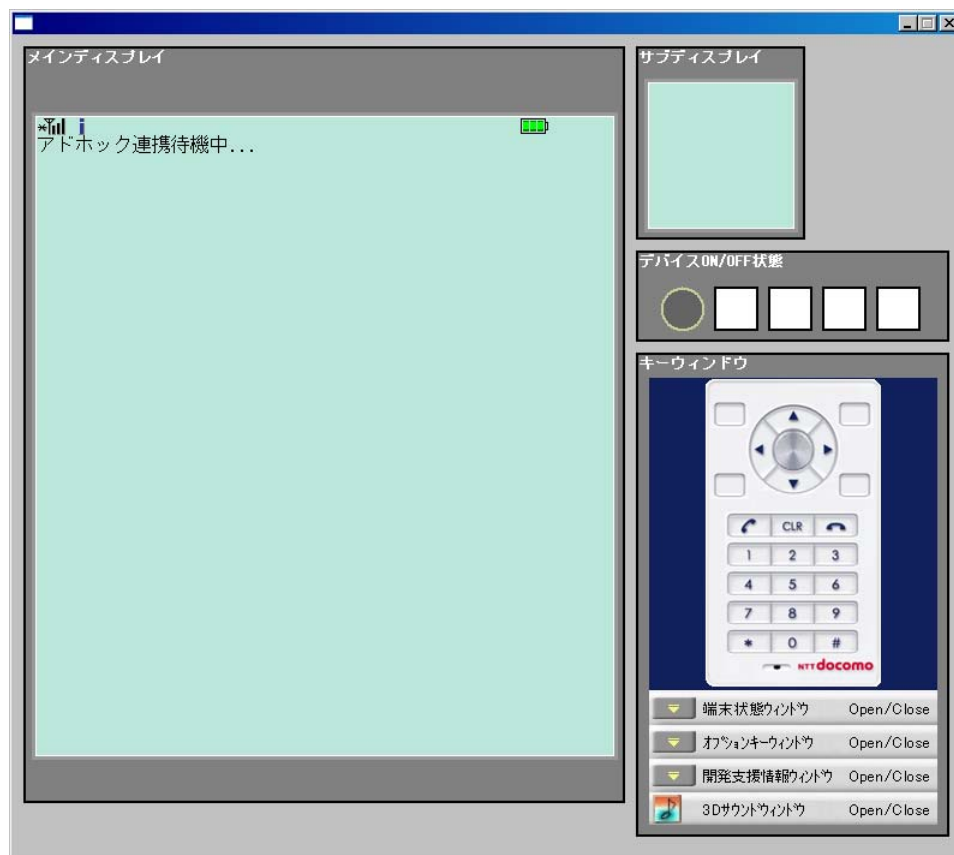


図 4-48 アドホック通信待機中画面

3. クライアントからの接続が確立すると「通信中」となり、StarEventListener#updateStarApplication(StarEventObject)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外部接続待ち」を選択します（図 4-49）。

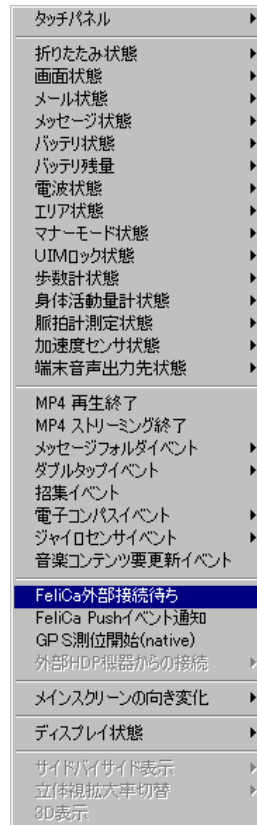


図 4-49 FeliCa 外部接続待ち

コンソール上に「FeliCa 外部接続待機開始しました」と表示され、「FeliCa 外部接続待機終了しました」と表示されるまでの間、相手からの接続を待ち受けます。

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

「iアプリTool」でi アプリをトルカから起動するには：

1. メイン画面の「起動種別」プルダウンから「トルカ起動」を選択し、「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 4-50）。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（「4.16 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」、「4.34 MyFACE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。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. DTV起動」を参照）が起動している。



図 4-50 トルカ選択画面

2. 選択したトルカのトルカ画面が表示されます（図 4-51）。
（この画面は実際のトルカとは異なります。）

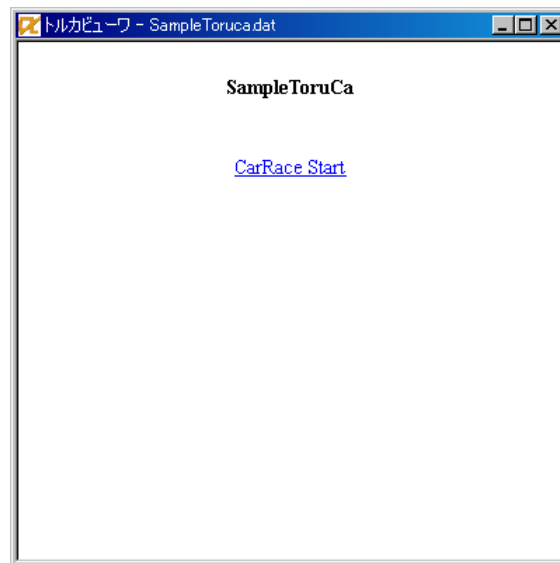


図 4-51 トルカ画面

3. 図 4-51 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（「4.8. 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（「15.15. 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:\¥iDKStar2.0¥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 4-52）、「はい」ボタンを押すと i アプリが起動します。

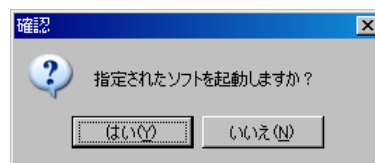


図 4-52 起動確認ダイアログ

トルカデータの作成については、（「7.8. トルカデータ編集」および「15.15. トルカファイル」）を参照してください。

4. i アプリを終了すると、再びトルカ画面が表示されます。
ただし、i アプリを強制終了（エミュレーション端末画面の「終話ボタン」を押して終了）した場合、トルカ画面は表示されません。

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、実際の携帯電話において 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（「4.4. 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

「iα ppliTool」で i アプリを削除時起動するには：

1. メイン画面の「起動種別」プルダウンから「削除時起動」を選択し、「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、DTV に対応した携帯電話からのみ利用可能です。

4.25.1. BML 起動手順

「i appliTool」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「起動種別」プルダウンから「BML 起動」を選択し、「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 4-53）。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（「4.16 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」、「4.34 MyFACE 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（「4.23. トルカ起動」を参照）が起動している。
- DTV（「4.26. DTV 起動」を参照）が起動している。



図 4-53 BML ブラウザ画面

2. 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 4-54）。

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 `StarApplicationManager.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。

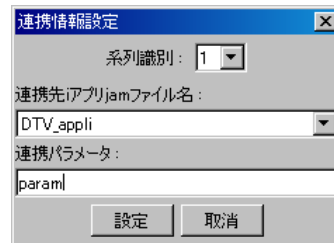


図 4-54 連携情報設定画面

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。BMLブラウザ画面（図 4-53）に戻ります。

3. BMLブラウザ画面（図 4-53）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」（「4.8」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（図 4-55）、「はい」ボタンを押すと i アプリが起動します。



図 4-55 起動確認ダイアログ

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 に対応した携帯電話からのみ利用可能です。

4.26.1. DTV 起動手順

「iアプリTool」で i アプリを DTV から起動するには：

1. メイン画面の「起動種別」プルダウンから「DTV 起動」を選択し、「起動」ボタンを押します。

DTV画面が表示されます（図 4-56）。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（「4.16 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」、「4.34 MyFACE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（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。

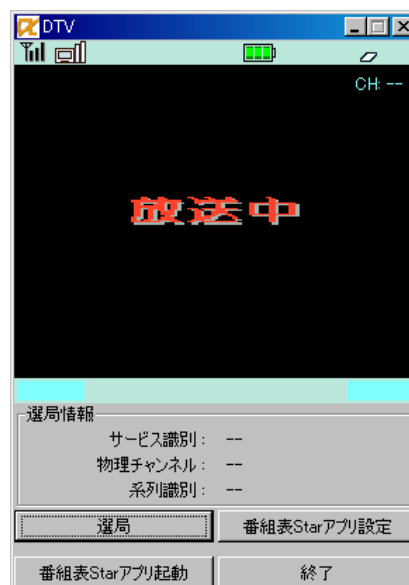


図 4-56 DTV 画面

2. 選局情報を設定したい場合は、DTV 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 4-57）。この画面では、DTV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 DTV 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 `StarApplicationManager.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 `NULL` が取得されます。

物理チャンネル：

表示中の DTV 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 DTV 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。

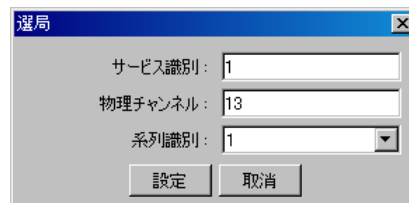


図 4-57 選局画面

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。DTV画面（図 4-56）に戻ります。

3. 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 Star アプリ設定」ボタンを押します。

番組表 i アプリ設定画面が表示されます（図 4-58）。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。

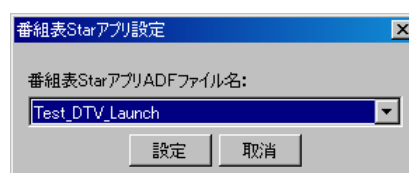


図 4-58 番組表 Star アプリ設定画面

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。DTV画面（図 4-56）に戻ります。

4. DTV画面（図 4-56）で「番組表Star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（「4.8. 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 アプリの許可」が Yes に設定されていること（「4.9. 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DTVから起動するための確認ダイアログが表示され（図 4-59）、「はい」ボタンを押すと i アプリが起動します。

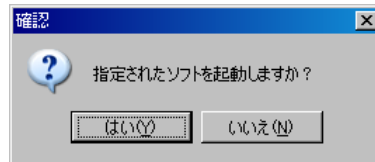


図 4-59 起動確認ダイアログ

4.27. 地図プラットフォーム起動/ナビ起動

「地図プラットフォーム起動」とは、ネイティブメニューから位置情報を渡して地図プラットフォームアプリを直接起動する方法で「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（「4.9. トラステッド動作の設定」参照）。
- 「地図プラットフォーム起動」の起動タイプは、「LAUNCHED_AS_MAP_PLATFORM_DIRECTLY」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」→「地図プラットフォームにおける電話帳内情報の利用の許可」を Yes にした場合のみ起動可能となります（「4.8. 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）。

「ナビ起動」とは、ネイティブメニューから位置情報を渡さずに地図プラットフォームアプリを直接起動する方法で「通常起動」と異なる点は「地図プラットフォーム起動」の場合と同様です。

4.27.1. 地図プラットフォーム起動/ナビ起動手順

「i alpha Tool」で i アプリを地図プラットフォームとして起動するには：

1. メイン画面の「起動種別」プルダウンから「地図プラットフォーム起動」または「ナビ起動」を選択し、「起動」ボタンを押します。
2. 地図プラットフォーム起動で i アプリが起動します。

「地図プラットフォーム起動」を選択した場合に渡される位置情報のパラメータは固定値で以下のとおりです：

パラメータ (StarApplicationManager.getParameter)	内容	値
"latitude"	緯度	N0.00.00.00
"longitude"	経度	E0.00.00.00
"datum"	測地系	0 : WGS84
"level"	測位レベル	49
"timestamp"	測位日時	2008/01/01 00:00:00 000

4.28. 招集メッセージ起動

「招集メッセージ起動」とは、i モード携帯電話に届いた招集メッセージ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（「4.9. トラステッド動作の設定」参照）。
- 「招集メッセージ起動」の起動タイプは、「LAUNCHED_BY_INVITE_MESSAGE」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」→「i アプリコールの利用(招集メッセージによる起動)の許可」を Yes にした場合のみ起動可能となります（「4.8. 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）。

招集メッセージは SMS として送信され、複数の相手の i モード携帯電話に起動要求を投げてアプリを起動させることができます。

4.28.1. 招集メッセージ起動手順

「i α ppliTool」で i アプリを招集メッセージから起動するには：

1. メイン画面の「起動種別」プルダウンから「招集メッセージ起動」を選択し、「起動」ボタンを押します。
2. 招集起動画面が表示されるので、パラメータを入力して「起動」ボタンを押します（図 4-60）。
ここでのパラメータは、プロジェクト設定画面でも設定できます（「4.6. プロジェクトの設定」参照）。

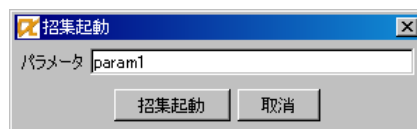


図 4-60 招集起動画面

3. 招集メッセージ起動で i アプリが起動します。

4.29. コピー文字列起動

「コピー文字列起動」とは、選択された任意の文字列から簡易に地図プラットフォーム i アプリや位置情報取得 i アプリを起動する方法で「通常起動」とは以下の点が異なります：

- 「コピー文字列起動」の起動タイプは「LAUNCHED_FROM_SELECTED_WORDS」です。
- ADFの設定項目「GetLocationParam」に「Yes」を設定した場合、または「トラステッド動作設定」→「位置情報APIの利用の許可」をYesにした場合に起動可能となります（「4.8. 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）。

4.29.1. コピー文字列起動手順

「iα ppliTool」で i アプリをコピー文字列から起動するには：

1. メイン画面の「起動種別」プルダウンから「コピー文字列起動」を選択し、「起動」ボタンを押します。
2. 文字列の入力画面が表示されるので、i アプリへ引き渡す文字列を入力します（図 4-61）。

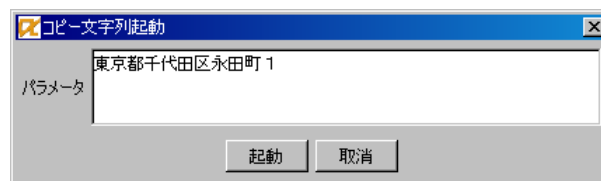


図 4-61 コピー文字列起動パラメータの入力画面

3. 「起動」ボタンを押します。
4. コピー文字列起動で i アプリが起動します。

コピー文字列起動パラメータの入力画面で指定した文字列は、i アプリから `StarApplicationManager.getParameter` の引数に “free_words” を指定することで取得できます。

4.30. スケジューラ連携起動

「スケジューラ連携起動」とは、i アプリを携帯電話のスケジューラ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：

- 「スケジューラ連携起動」の起動タイプは「LAUNCHED_FROM_SCHEDULER」です。
- ADFの設定項目「LaunchBySchedule」に“**Yes**”を設定した場合に起動可能となります（「4.8. 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）。

4.30.1. スケジューラ連携起動手順

「iα ppliTool」でi アプリをスケジューラから起動するには：

1. メイン画面の「起動種別」プルダウンから「スケジューラ起動」を選択し、「起動」ボタンを押します。
2. スケジューラ起動画面が表示されるので、パラメータを入力して「起動」ボタンを押します（図 4-62）。
パラメータは「名前=値」の形式で指定します。複数のパラメータを指定する場合は、「名前=値, 名前=値,...」のように「,」で区切ります。「名前」と「値」の文字列長の合計が 20480 バイト以内で指定可能です。

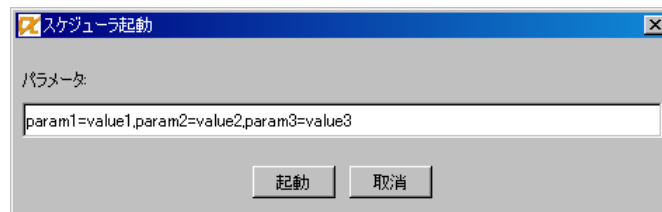


図 4-62 スケジューラ起動画面

3. スケジューラ起動でi アプリが起動します。
スケジューラ起動画面で指定した文字列は、i アプリから
StarApplicationManager.getParameter の引数に指定したパラメータ名を指定することで取得できます。

4.31. データ BOX 画像連携起動

「データ BOX 画像連携起動」とは、データ BOX の画像（静止画）から画像利用アプリを起動する方法で「通常起動」とは以下の点が異なります：

- 「データ BOX 画像連携起動」の起動タイプは「LAUNCHED_FROM_IMAGE」です。
- ADFの設定項目「LaunchByDataBox」に“picture”を設定した場合に起動可能となります（「4.8. 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）。

4.31.1. データ BOX 画像連携起動手順

「iαpliTool」で i アプリをデータ BOX の画像から起動するには：

1. メイン画面の「起動種別」プルダウンから「データ BOX 画像起動」を選択し、「起動」ボタンを押します。
2. データBOX画像起動画面が表示されるので、起動元の画像ファイル名（フルパス）を入力して「起動」ボタンを押します（図 4-63）。画像ファイル名は直接入力することも、「参照」ボタンを押して画像ファイルの選択画面から選択することもできます（図 4-64）。
「データBOX画像連携起動」で指定できる画像ファイルは、マルチメディア領域の「マイピクチャ」に作成・管理されている画像ファイルのみです。

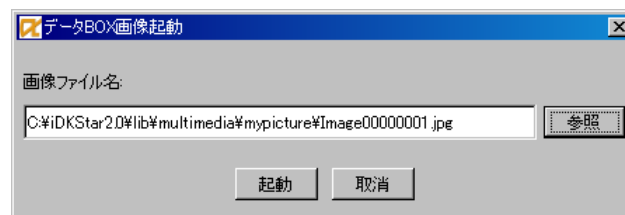


図 4-63 データ BOX 画像起動画面

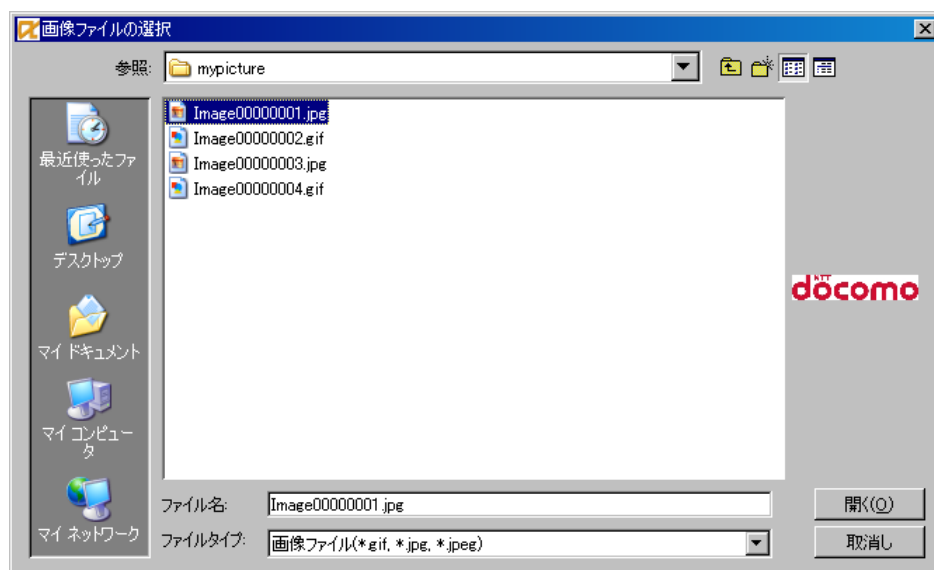


図 4-64 画像ファイルの選択画面

3. データ BOX 画像連携起動で i アプリが起動します。
起動した i アプリから、StarApplicationManager.getParameter の引数に “entry-id” を指定することで、起動元の画像のエントリ ID を取得できます。このエントリ ID を ImageStore.getEntry(int) の引数に指定することで、起動元の画像 (ImageStore オブジェクト) を取得できます。

4.32. Bluetooth 連携起動

「Bluetooth 連携起動」とは、i アプリを外部 Bluetooth 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：

- 「Bluetooth 連携起動」の起動タイプは「LAUNCHED_BY_BLUETOOTH_REMOTE_DEVICE」です。
- ADF の「AllowPushBy」に「bt:」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 Bluetooth SPP 通信を仮想的に実現する事ができます。また、Bluetooth HDP 機器との通信を、データファイルの読み込み・書き込みによって仮想的に実現することができます。

Bluetooth通信を実行する上での通信の設定については、「6.6 Bluetooth設定」を参照してください。

4.32.1. Bluetooth 連携起動手順

「iα ppliTool」で i アプリを Bluetooth 連携起動するには：

1. メイン画面の「起動種別」プルダウンから「Bluetooth 起動」を選択し、「起動」ボタンを押します。
2. Bluetooth接続要求待機中になり、StarApplicationManager#addRemoteLauncher()で事前に登録した接続先クライアントからの接続要求をPage Scanモードで待ち受けます（図 4-65）。

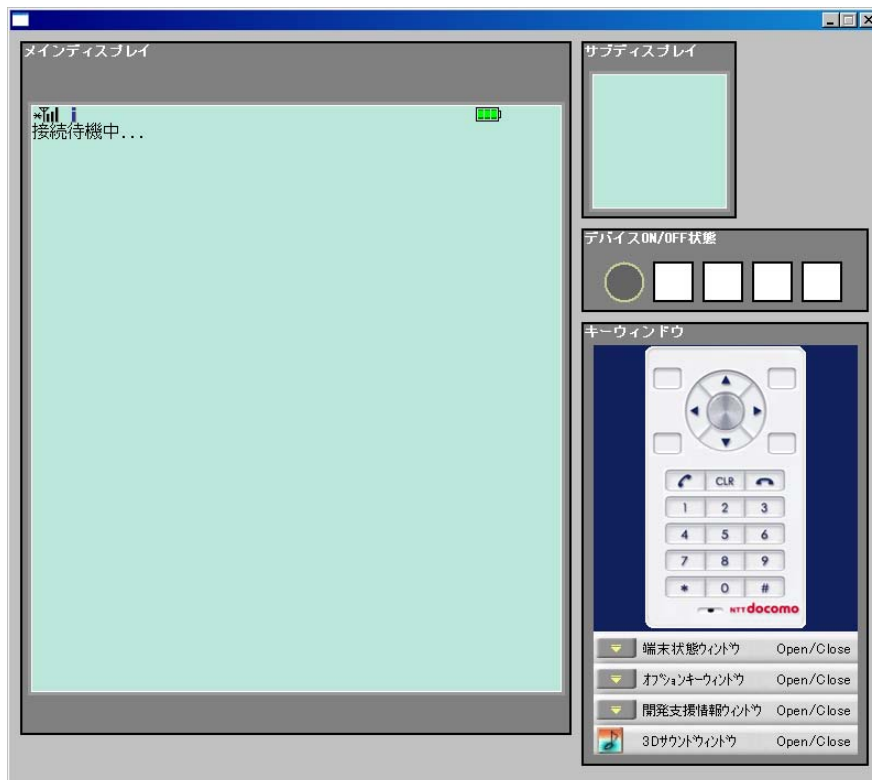


図 4-65 Bluetooth 接続要求待機中画面

3. 対向となるエミュレータで Bluetooth アプリを起動中の場合は、対向 Bluetooth アプリ側で Page 処理 (Bluetooth#searchAndSelectDevice()、または Bluetooth#startPairing()) を行うことで接続が確立し、アプリが起動されます。

4.32.2. Bluetooth HDP 機器からの接続要求による連携起動

HDP機器からの接続要求をエミュレートする場合は、図 4-65 の接続要求待機中画面で、エミュレーション端末の右クリックメニューから「外部HDP機器からの接続」を選択してください。



図 4-66 Bluetooth 外部 HDP 機器選択画面

ここで表示される外部 HDP 機器は、StarApplicationManager#addRemoteLauncher() で事前に登録されている機器のみです。
ここで外部機器を選択し「確定」をクリックすることで、選択した HDP 機器との接続が確立したものととしてアプリが起動されます。

4.33. パッケージ起動

「パッケージ起動」とは、コンテンツパッケージから i アプリを起動する方法で「通常起動」とは以下の点が異なります：

- 「パッケージ起動」の起動タイプは「LAUNCHED_FROM_CONTENTS_PACKAGE」です。

4.33.1. パッケージ起動手順

「iα ppliTool」で i アプリをパッケージ起動するには：

1. メイン画面の「起動種別」プルダウンから「パッケージ起動」を選択し、「起動」ボタンを押します。
2. パッケージ起動画面が表示されるので、パラメータを入力して「起動」ボタンを押します（図 4-67）。
パラメータは「名前=値」の形式で指定します。複数のパラメータを指定する場合は、「名前=値, 名前=値,...」のように「,」で区切ります。「名前」と「値」の文字列長の合計が 20480 バイト以内で指定可能です。

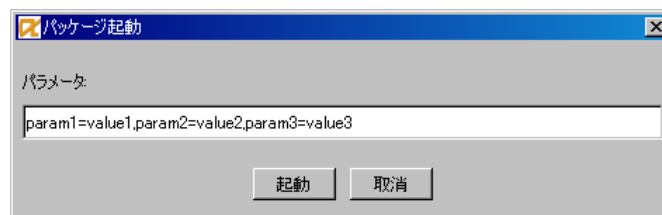


図 4-67 パッケージ起動画面

3. パッケージ起動で i アプリが起動します。
パッケージ起動画面で指定した文字列は、i アプリから `StarApplicationManager.getParameter` の引数に指定したパラメータ名を指定することで取得できます。

4.34. MyFACE 起動

「MyFACE 起動」とは、i アプリを MyFACE から起動することです。

「MyFACE 起動」の起動タイプは、「LAUNCHED_FROM_FACE」です。

「iα ppliTool」では、MyFACE機能は搭載していませんが、「ブラウザ起動」で使用する簡易 i モードブラウザを使用し、HTMLファイル（MyFACEコンテンツ本体のインデックスファイル）から i アプリを連携起動することでMyFACE起動をエミュレートします。

4.34.1. MyFACE 起動手順

「iα ppliTool」で i アプリを MyFACE 起動するには：

1. メイン画面の「起動種別」プルダウンから「MyFACE起動」を選択し、「起動」ボタンを押します。

MyFACE起動用のブラウザ画面が表示されます（図 4-68）。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（「4.16 メーラ起動」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（「4.23. トルカ起動」を参照）が起動している。
- BMLブラウザ（「4.25. BML起動」を参照）が起動している。
- DTV（「4.26. DTV起動」を参照）が起動している。

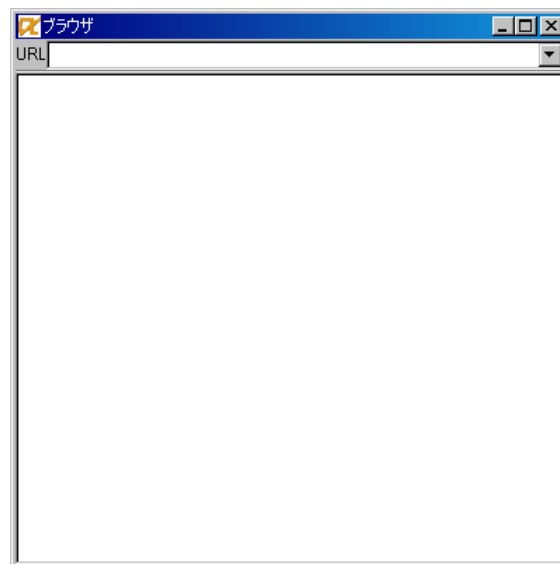


図 4-68 ブラウザ画面

2. ブラウザに表示するHTMLファイル（MyFACEコンテンツ本体のインデックスファイル）を編集して起動するアプリを追記します（「15.4. ブラウザ起動HTMLファイル」参照）。

3. 図 4-68 の上方のURLフィールドに 2 で編集したHTMLファイル（MyFACEコンテンツ本体のインデックスファイル）のURLを入力し、リターンキーを押します（図 4-69）。

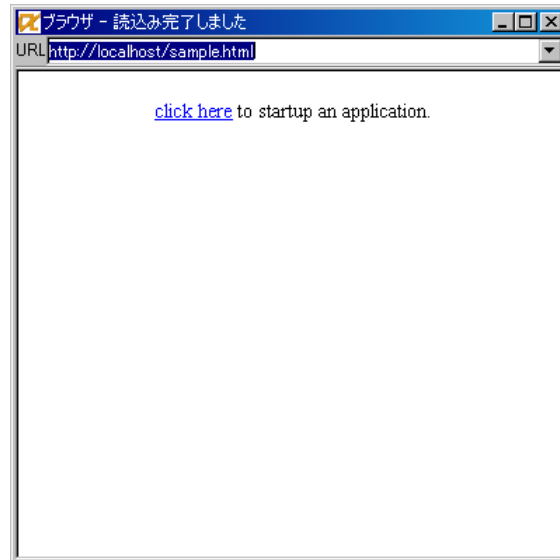


図 4-69 URL の表示

4. 図 4-69 のリンク（青い文字）をクリックします。

i アプリを MyFACE 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「LaunchByFaceApp」に「any」あるいは起動を許可するURLが設定されていること（「4.8. 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（MyFACEコンテンツ本体のインデックスファイル）に起動するアプリが正しく指定されていること（「15.4. ブラウザ起動HTML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:\¥iDKStar2.0¥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MyFACE起動するための確認ダイアログが表示され（図 4-70）、「はい」ボタンを押すと i アプリが起動します。

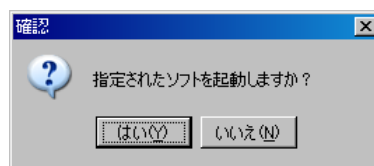


図 4-70 起動確認ダイアログ

5. i アプリを終了すると、再びブラウザ画面が表示されます。
ただし、i アプリを強制終了（エミュレーション端末画面の「終話ボタン」を押して終了）した場合、ブラウザ画面は表示されません。

4.35. Widget View 表示

「Widget View 表示」とは、「iα ppliTool」画面の「Widget View 表示ボタン」を押したときに、Widget View を起動させることです。

この時、Widget View のランチャー画面に表示されるアプリケーションは、プロジェクト読み込み時に指定したプロジェクト格納先にあるミニアプリ、ミニ+フルアプリ、フル+ミニアプリが表示されます。（デフォルトは、C:\¥iDKStar2.0¥apps 配下です。）



図 4-71 Widget View 表示

Widget View 表示では、ランチャー表示画面でミニアプリを選択する事により、ミニアプリを複数起動することができます。

これにより、ミニアプリにおける Focused 状態 / Unfocused 状態 / Selected 状態 / Unselected 状態 の動作を試験することができます。

ミニアプリは、最大 9 件まで起動可能で、8 件まで貼り付けられます。

9 件目を起動後、一覧表示画面に遷移する際に 9 件目のミニアプリの停止を要求します（図 4-72）。



図 4-72 ミニアプリ停止要求

Widget View のランチャー表示画面にある「全てのアプリ」を選択した場合、実際はアプリケーション一覧が表示されるのですが、エミュレータでは、Widget View を停止させます。

4.36. Widget View 起動

「Widget View 起動」とは、何らかの動作により Widget View が停止した場合、Widget View を再び起動したときに、前回、Widget View 上で動いていた(貼りついていた)ミニアプリを再起動(復元)することです。

エミュレータでは、ミニアプリの貼り付け状態は保存しないため、前回起動されていたミニアプリを復元することはありません。

そこで、現在読み込まれているアプリケーションがミニアプリ、フル+ミニアプリ、ミニ+フルアプリである場合において、Widget View 表示ボタンを押した際に「Widget View 起動」で起動します。

このときの起動タイプは、「LAUNCHED_FROM_WIDGETVIEW」になります。



図 4-73 Widget View 起動

4.37. 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
- 初回起動
- タイマ起動
- 外部機器起動
- 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
- DTV 起動
- 招集メッセージ起動
- 地図プラットフォーム起動/ナビ起動
- コピー文字列起動
- スケジューラ連携起動
- データ BOX 画像連携起動
- Bluetooth 連携起動
- パッケージ起動
- MyFACE 起動
- Widget View 起動

ここで、フルアプリとミニアプリおよび、これらを組み合わせた場合の画面遷移を図 4-74 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
- DTV が起動していない



図 4-74 「iα ppliTool」における画面遷移

※ フルアプリとミニアプリの切り替えについては、「第 13 章. フルミニ連携」を参照してください。

第5章 iアプリ Bookmark の作成

この章では、i アプリ Bookmark（コンテンツパッケージ（CP）に含めることができる i アプリ）の作成について説明します。

i アプリ Bookmark は、i アプリのダウンロード元ページの情報を持つリンクです。
i モード携帯電話上では、i アプリ一覧から i アプリ Bookmark を選択することで、対象の i アプリをダウンロードすることができます。
（Widget View 上では表示されません）

「i α ppli Development Kit for Star-2.0」では、i アプリ Bookmark を作成する事ができます。
（i アプリ Bookmark の実行には対応しておりません）

i アプリ Bookmark を i モード携帯電話で動作させるには、次のものが必要になります。

- i アプリ Bookmark の JAR ファイル
（i アプリ一覧に表示するアイコン画像・サムネイル画像・メインタイトル画像を利用する場合のみ）
- アプリケーションディスクリプタファイル（ADF）

これらは、すべて「i α ppli Development Kit for Star-2.0」で簡単に作成することができます。

5.1. iアプリ Bookmark の作成手順

「iα ppliTool」では、iアプリと同様に i アプリ Bookmark を「プロジェクト」単位で作成します。

「iα ppliTool」では、i アプリ Bookmark の作成を次の手順で行います：

- ① 新しいプロジェクトの作成（「5.2. プロジェクトの作成」を参照）
- ② 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（「5.3 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）
- ③ プロジェクトのビルド（5.4プロジェクトのビルド」を参照）

なお、i アプリ Bookmark では、スクラッチパッドは利用しません。

5.2. プロジェクトの作成

「iα ppliTool」では、以下の2通りの方法でiアプリBookmarkを作成することができます。

- プロジェクト新規作成
プロジェクト新規作成より、iアプリBookmarkプロジェクトを新規に作成します。
本方法では、対応する（ダウンロード対象の）iアプリに関する情報（ADF項目）を全て設定していただく必要があります。
- 既存のiアプリプロジェクトから作成
対応する（ダウンロード対象の）iアプリ情報を引き継いで、iアプリBookmarkプロジェクトを新規に作成します。
本方法では、対応する（ダウンロード対象の）iアプリに関する基本的な情報（ADF項目）が既に設定された状態でiアプリBookmarkの作成を始めることができます。

それぞれの作成方法について以下に説明します。

5.2.1. プロジェクト新規作成

新規にiアプリBookmark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 5-1）。

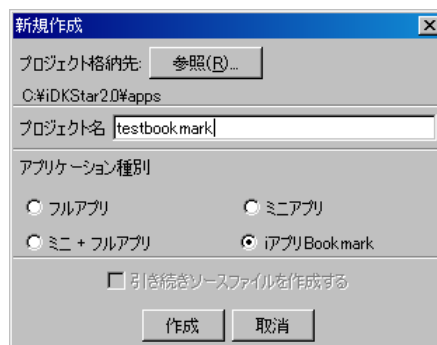


図 5-1 プロジェクト新規作成画面

2. プロジェクトの格納先（C:\iDKStar2.0\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ます。
4. アプリケーション種別でiアプリBookmarkを選択します。

5. 「作成」ボタンを押します。^[8]

i アプリ Bookmark 選択時は、「引き続きソースファイルを作成する」のチェックは使用できません。

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）の格納場所を示すメッセージが表示されます（図 5-2）。

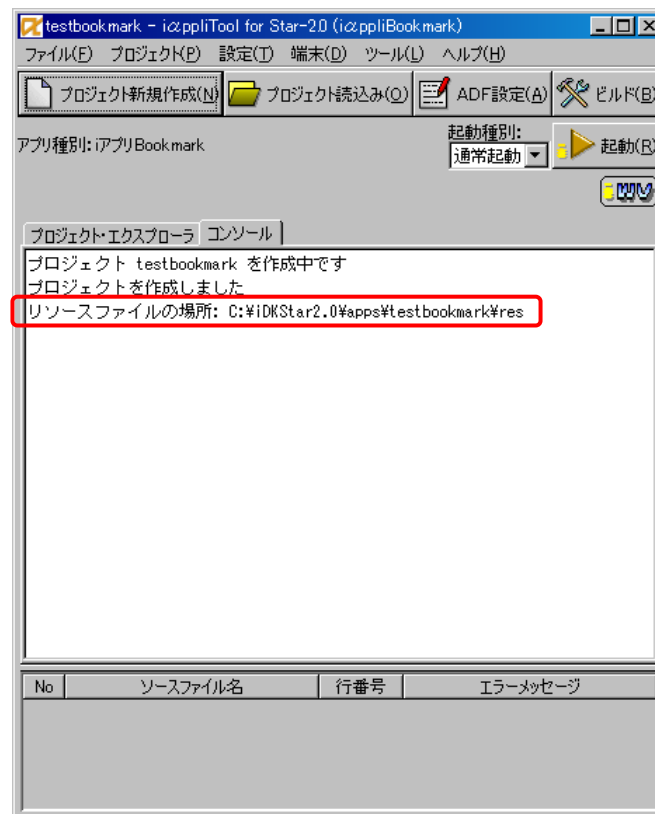


図 5-2 新規プロジェクト作成後のメイン画面

^[8] i アプリ Bookmark の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：

リソースファイルの格納場所: C:\%iDKStar2.0%\apps<プロジェクト名>%res
JAR、ADF の格納場所: C:\%iDKStar2.0%\apps<プロジェクト名>%bin

5.2.2. 既存のiアプリプロジェクトから作成

対応する i アプリのプロジェクトの情報を用いて新規に i アプリ Bookmark プロジェクトを作成するには：

1. 対応する 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Bookmark作成...」を選択します（図 5-3）。

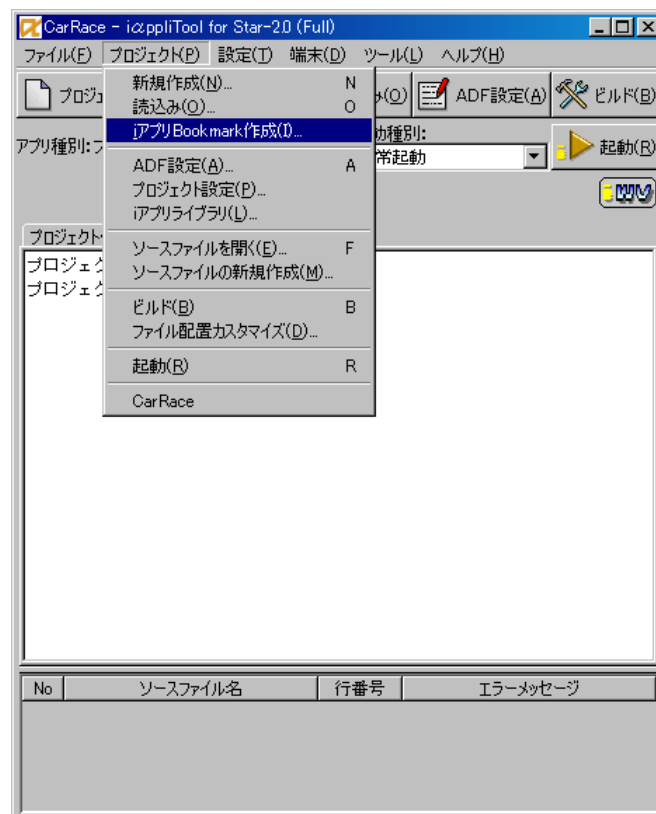


図 5-3 i アプリ Bookmark 作成メニュー

「i アプリBookmark新規作成画面」が表示されます（図 5-4）。



図 5-4 i アプリ Bookmark 新規作成画面

3. プロジェクトの格納先（C:¥iDKStar2.0¥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。
4. プロジェクト名を入力します。

5. 「作成」ボタンを押します。
(この時作成されるディレクトリは「5.2.1 プロジェクト新規作成」と同様です)

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）の格納場所を示すメッセージが表示されます（図 5-5）。

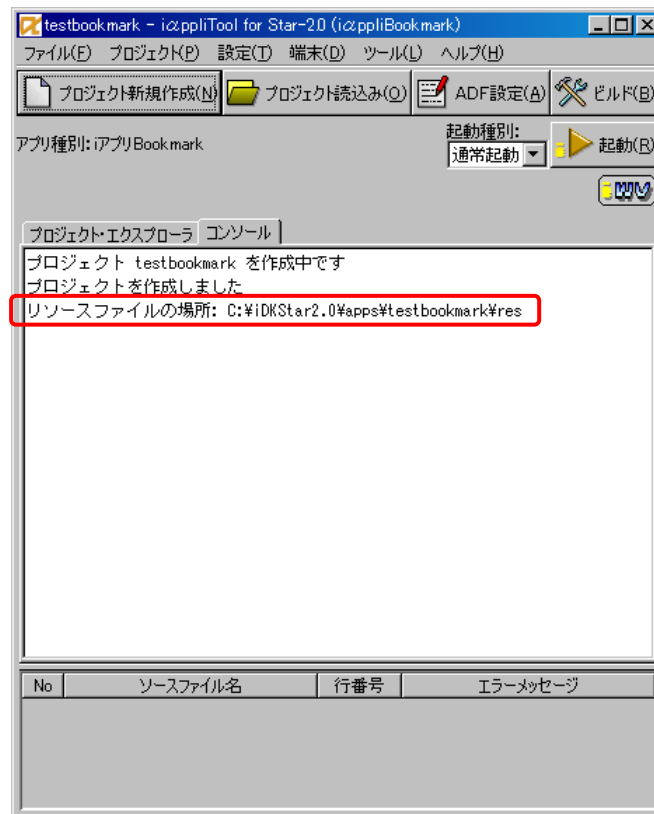


図 5-5 新規プロジェクト作成後のメイン画面

5.3. アプリケーションディスクリプタファイル(ADF)の編集

i アプリ Bookmark についても、i アプリと同様に Java Application Manager (JAM) で実行を管理しています。

そのため、JAR ファイルとは別に i アプリ Bookmark 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。

設定方法は i アプリと同様ですが、i アプリ Bookmark は i アプリに比べ設定可能な情報が少なく、i アプリ Bookmark 専用の項目および i アプリと共通の項目 (一部) で構成されています。

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 5-6)。

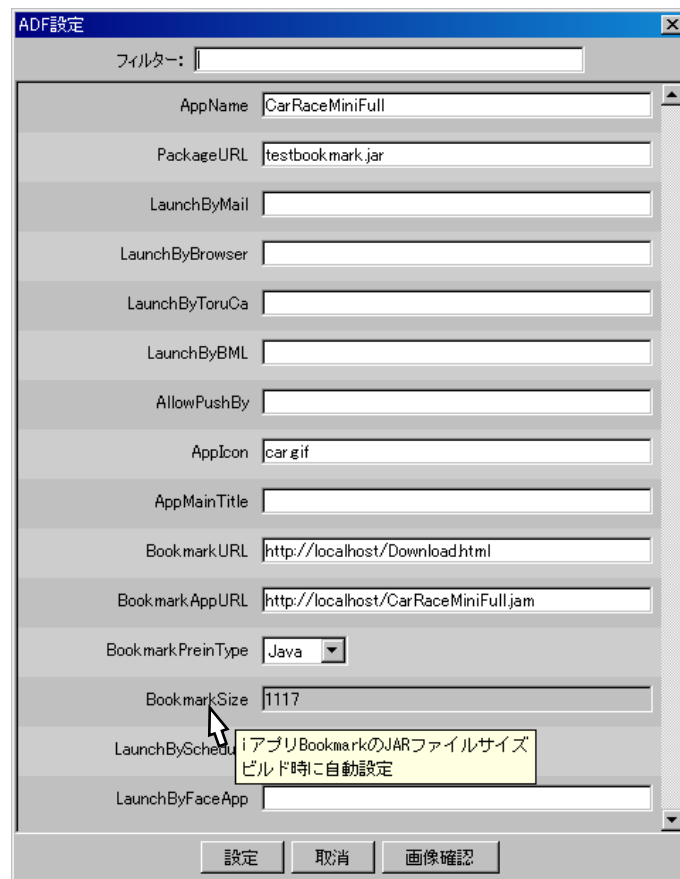


図 5-6 ADF 設定画面

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。
以降に各設定項目の内容を説明します。

AppName

対象	説明	種別
i アプリ Bookmark	i アプリ Bookmark の表示名称（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、プロジェクト名が設定されます。	必須

PackageURL

対象	説明	種別
i アプリ Bookmark	i アプリ Bookmark の JAR パッケージライブラ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。 ----- 新規プロジェクトを作成したときには、i アプリ Bookmark の JAR ファイル名になっているので、 <u>適切な URL（i アプリ Bookmark のダウンロード先）に変更してください。</u>	オプション （コンテンツパッケージダウンロード機能では必須）

LaunchByMail

- i アプリと同様です。
- i アプリの LaunchByMail の項目を参照してください。

LaunchByBrowser

- i アプリと同様です。
- i アプリの LaunchByBrowser の項目を参照してください。

LaunchByToruCa

- i アプリと同様です。
- i アプリの LaunchByToruCa の項目を参照してください。

LaunchByBML

- i アプリと同様です。
- i アプリの LaunchByBML の項目を参照してください。

AllowPushBy

- i アプリと同様です。
- i アプリの AllowPushBy の項目を参照してください。

AppIcon

- i アプリと同様です。
- i アプリの AppIcon の項目を参照してください。

AppMainTitle

- i アプリと同様です。
- i アプリの AppMainTitle の項目を参照してください。

BookmarkURL

対象	説明	種別
i アプリ Bookmark	i アプリ Bookmark から接続するサイトの URL（最大 255 バイト）。 設定する文字列は、ASCII 形式とします。 絶対 URL のみ許容し、相対 URL を設定した場合は ADF チェック時にエラーとなります。（実機ではダウンロード不可となります。）	必須

BookmarkAppURL

対象	説明	種別
i アプリ Bookmark	i アプリ Bookmark に対応するアプリの、JAR パッケージライブラリの格納場所を表す URL（最大 2048 バイト）。 設定する文字列は、ASCII 形式とします。 i アプリ Bookmark への連携起動時においては、このキーで指定された値を i アプリ Bookmark の ADF の URL として扱います。	必須

BookmarkPreinType

対象	説明	種別
i アプリ Bookmark	i アプリ Bookmark の対象が、Java アプリか、IC アプリかを示す文字列。 設定する文字列は、ASCII 形式で、“ Java ”（Java アプリ）、“ JavaIC ”（IC アプリ）とし、大文字・小文字を区別します。	必須

BookmarkSize

対象	説明	種別
i アプリ Bookmark	i アプリ Bookmark の JAR ファイルのサイズ（バイト単位）。 【サイズ制限】 ●100KByte まで。 ----- 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。 ビルド時に JAR ファイルが生成されない場合（格納ファイルがない場合）は、本項目は設定されません。	オプション （コンテンツパッケージダウンロード機能では必須）

LaunchBySchedule

i アプリと同様です。

i アプリの LaunchBySchedule の項目を参照してください。

LaunchByFaceApp

i アプリと同様です。

i アプリの LaunchByFaceApp の項目を参照してください。

このアプリケーションディスクリプタファイル（ADF）は、
「C:\iDKStar2.0¥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。

5.4. プロジェクトのビルド

i アプリ一覧に表示するアイコン画像・サムネイル画像・メインタイトル画像を利用する場合は、ビルド前にあらかじめリソースファイルの格納場所に画像ファイルを格納しておく必要があります。

i アプリ Bookmark プロジェクトのビルドを行うには：

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 5-7）。

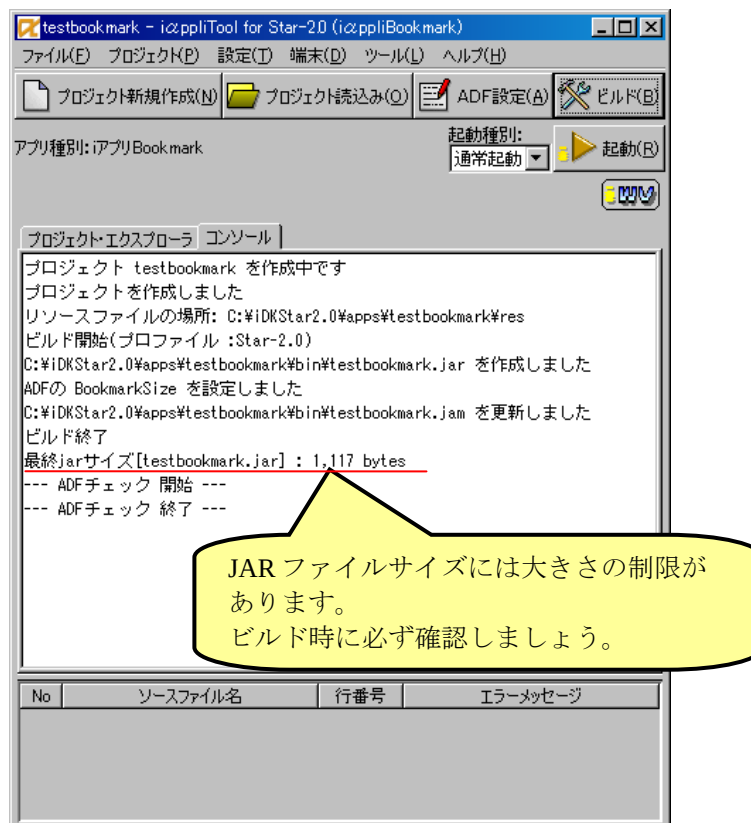


図 5-7 プロジェクトのビルド

アプリケーションディスクリプタファイルの更新および必要に応じて JAR ファイルの作成を行います。

このとき、ADF の AppName が未設定だった場合、i アプリ Bookmark のプロジェクト名が設定されます。

JAR ファイルは、i アプリ一覧に表示するアイコン画像・サムネイル画像・メインタイトル画像のうち、少なくとも 1 ファイルが格納されている場合に作成されます。

1 ファイルも格納されていない場合は、JAR ファイルは作成されません。

第6章 アプリケーション動作環境設定

この章では、i モード携帯電話の通信設定など、i アプリ全般に関する機能の設定について説明します。

この章で説明する内容は、メニューから「設定」→「アプリケーション動作環境設定...」を選択したときの各種機能です。

6.1. 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 6-1）。

アプリケーション動作環境設定

仮想FeliCa設定 | Bluetooth設定 | Bluetooth HDP設定 | ハイパーマルチ機能設定

全般 | UART用COMポート設定 | ネットワーク設定 | サウンド設定 | メールアドレス設定

ADFのURL

UserAgent

HTTP設定

プロキシホスト

プロキシポート

HTTPS設定

プロキシホスト

プロキシポート

OBEX設定

接続先

設定 取消 適用

図 6-1 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 6-1 に示すとおりです。

表 6-1 ネットワークの設定項目

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 STAR_INET_CLIENT(c100; SJ)
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。「6.1.1. 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。「6.1.1. 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。 「6.1.1. 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

設定項目		説明	デフォルト
	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。 「6.1.1. 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
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。 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

6.1.1. 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、「iα ppliTool」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 6-1 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 16 章. 証明書の管理」を参照してください。

6.2. 全般設定

全般タブでは、i モード携帯電話上での全体の機能に関連する値を設定します。
ここでは、以下の項目を設定することが可能です。

- ヒープサイズ
- タイムゾーン
- 対応フォントサイズ
- 外部メモリ
- 内部ストレージ

アプリケーション動作全般の設定をするには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→[全般]を選択します（図 6-2）。

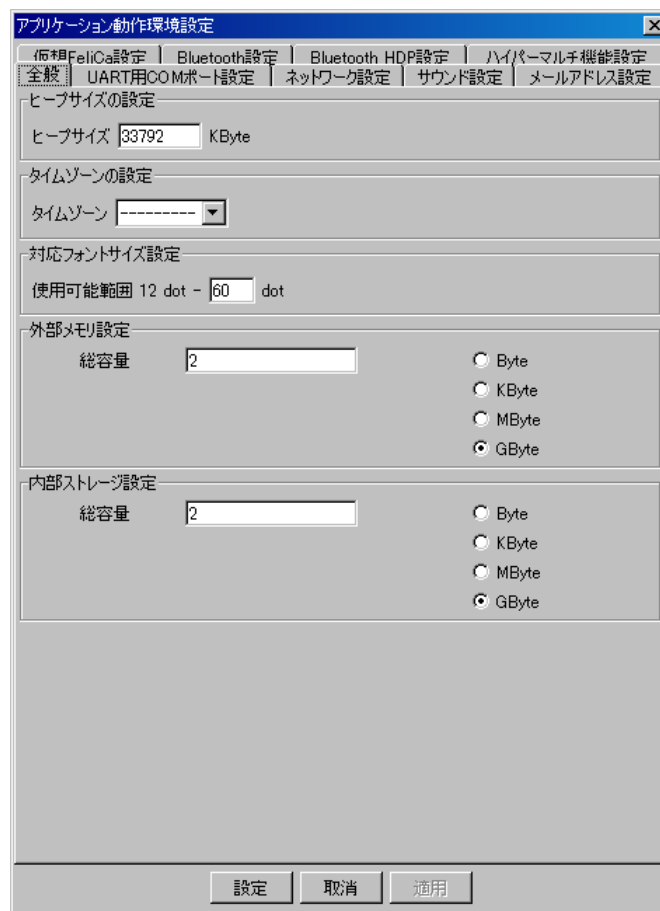


図 6-2 全般設定

2. 各設定項目を設定します。
各設定項目は、表 6-2 に示すとおりです。

表 6-2 全般設定の項目

設定項目		説明
ヒープサイズ		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 33792KByte 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。) ヒープサイズは、1000 ～ 65535 KByte の範囲で設定できます。
タイムゾーン		i モード携帯電話上のタイムゾーンを指定します。 「-----」を選択すると OS に依存します。デフォルトは OS 依存の設定になっています。 ここで指定したタイムゾーンは、 CallRecord#getString()、Mail#getString()、 Message#getString() に関連します。
対応フォントサイズ		i モード携帯電話上の対応フォントサイズの範囲（上限）を指定します。 ここで指定したフォントサイズは、Font.getFont() で取得でき、各種コンポーネントに設定する事ができます（指定した範囲内で 1dot 単位で取得可能です）。 対応フォントサイズの上限は、60 ～ 128 の範囲で設定できます（下限フォントサイズは 12dot 固定です）。
外部メモリ 設定	総容量	外部メモリ（ストレージデバイス）の総容量を指定します。 単位は Byte、KByte、MByte、GByte から選択可能ですが、いずれの場合でも 1024GByte 相当以下のサイズを指定してください。また仮想ストレージデバイスファイルパス (C:\iDKStar2.0\lib\storage\device) が存在するディスクドライブ (C ドライブ) の空き容量を超えるサイズは指定できません。 ストレージデバイスファイルパスに存在するファイルサイズの合計が、ここで設定する総容量を超えた場合、i アプリからファイル作成を試みると FileSystemFullException 例外が発生します。また Folder API で取得できる空き容量は、ここで設定する総容量からストレージデバイスファイルパスに存在するファイルサイズの合計を差し引いたものです。

設定項目		説明
内部ストレージ設定	総容量	内部ストレージとして使用可能な内部メモリ（DataBox 用メモリ）の総容量を指定します。 単位は Byte、KByte、MByte、GByte から選択可能ですが、いずれの場合でも 1024GByte 相当以下のサイズを指定してください。また仮想内部ストレージデバイスファイルパス（C:\iDKStar2.0\lib\internalstorage\device）が存在するディスクドライブ（C ドライブ）の空き容量を超えるサイズは指定できません。 内部ストレージデバイスファイルパスに存在するファイルサイズの合計が、ここで設定する総容量を超えた場合、i アプリからファイル作成を試みると FileSystemFullException 例外が発生します。また内部ストレージ API で取得できる空き容量は、ここで設定する総容量から内部ストレージデバイスファイルパスに存在するファイルサイズの合計を差し引いたものです。

3. 「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。

6.3. メールアドレスの設定

携帯電話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 6-3）。



図 6-3 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 6-4）。

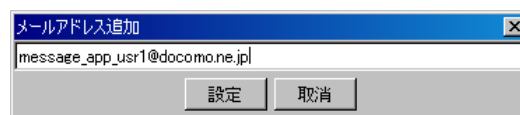


図 6-4 メールアドレス追加画面

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。

（「7.4.2. メールの送受信エミュレート機能」を参照）

6.4. 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 MFi 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 6-5）。

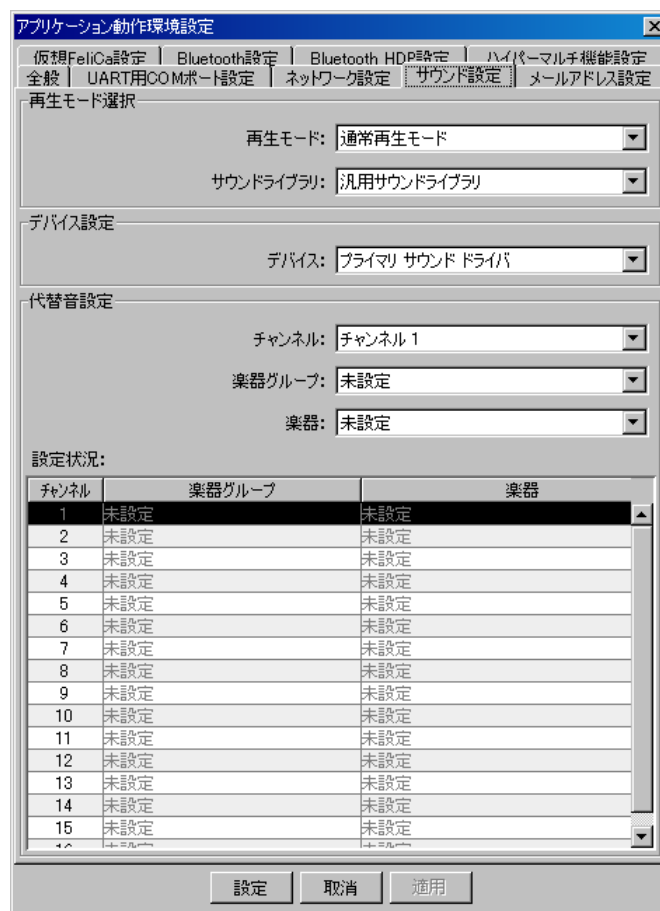


図 6-5 サウンド設定画面

2. 各設定項目を設定します。
各設定項目は、表 6-3 に示すとおりです。

表 6-3 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 PC 上で実際に 3D サウンドを再生することができます。ただし、お使いの PC のサウンドカードが 3D 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 3D サウンドは鳴らせませんが、3D サウンドを視覚的に表示する事ができます (8.16.4. 3D 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、PC 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 以降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。 PC 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 6-6)。 MFi 中の該当チャンネルで、PC 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。

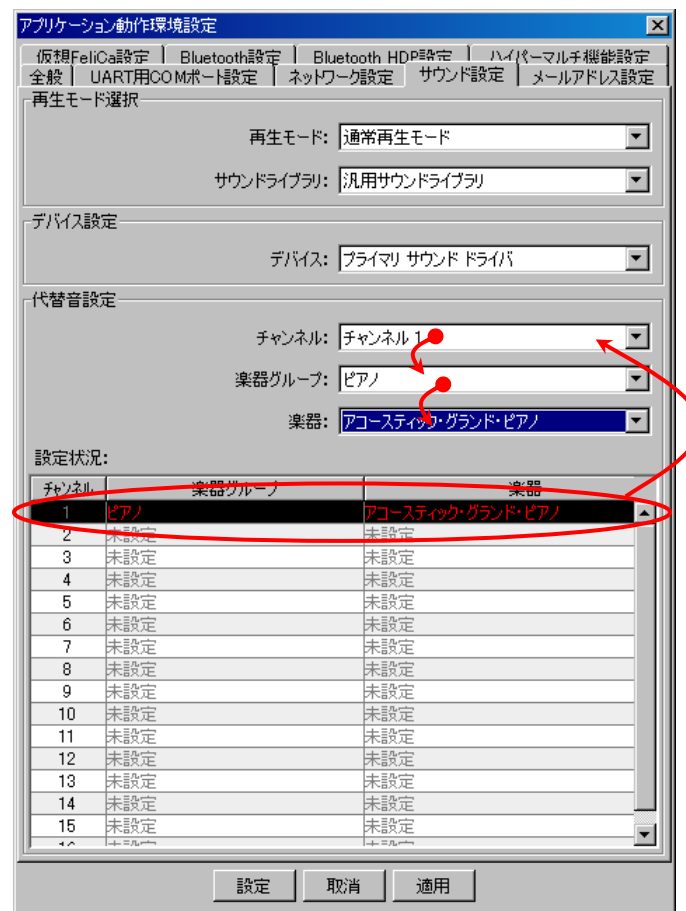


図 6-6 代替音設定

6.5. 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 PC 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 クラスの open メソッドで以下のように指定します。

<スキーム>;<パラメータ 1>=<値>,<パラメータ 2>=<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。 N0: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=N0,stopbitlen=1,flowctrl=OFF

※ UART の機能は携帯電話によって使えませんので、ターゲットとする携帯電話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 6-7）

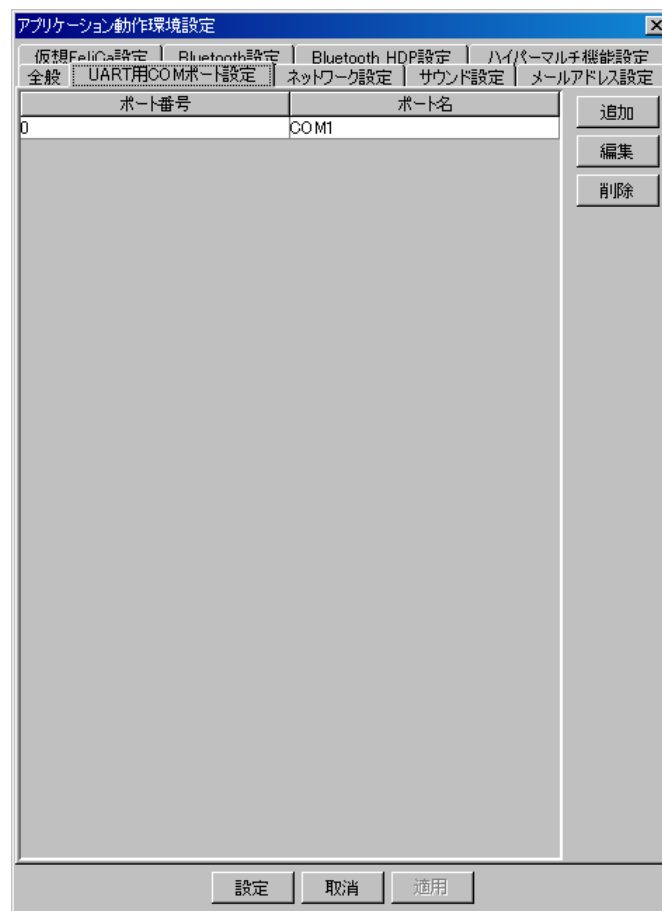


図 6-7 UART 用 COM ポート設定画面

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 6-8）。



図 6-8 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。

6.6. 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 SPPのAPIを利用した i アプリで、TCP/IPネットワーク上に接続された別のPCで起動しているエミュレータと i アプリ間のBluetooth通信をエミュレートする事ができます。またBluetooth HDPのAPIを利用した i アプリでは、ローカルファイル入出力によってBluetooth HDP機器との通信 (InputStream / OutputStream) をエミュレートすることができます (図 6-9)。

※ 実際の携帯電話では、Bluetooth 搭載機のみ扱えます。

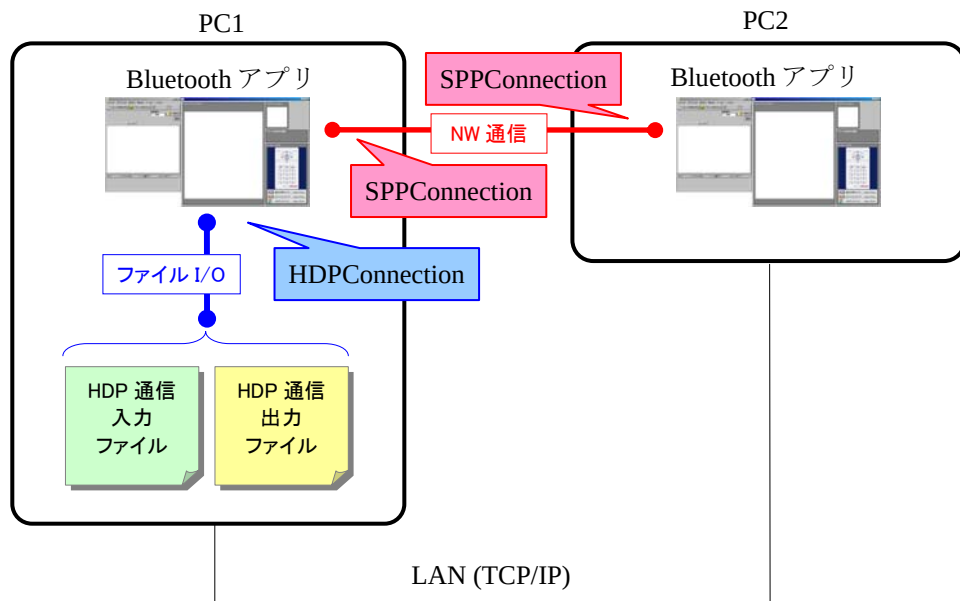


図 6-9 Bluetooth 通信のエミュレーション

6.6.1. Bluetooth 設定

Bluetooth 接続に関する基本的な設定を行います。

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 6-10)。

図 6-10 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 6-4 に示すとおりです。

表 6-4 Bluetooth の設定項目

設定項目	説明	デフォルト
自 Bluetooth デバイス設定 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。	StarBluetooth
自 Bluetooth デバイス設定 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 デバイス設定 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
自 Bluetooth デバイス設定 Spec Data	Extended Inquiry Response に含まれてい る Manufacturer Specific Data を設定しま す。 この情報は相手側に登録されます。	123456
自 Bluetooth デバイス設定 Local Name	Extended Inquiry Response に含まれてい る Shortened Local Name を選択します。 この情報は相手側に登録されます。	LocalName

設定項目	説明	デフォルト
自 Bluetooth デバイス UUID	Extended Inquiry Response に含まれている 16-bit Service Class UUIDs を選択します（エミュレータでは 1 件のみ登録できます）。 この情報は相手側に登録されます。	1234
登録機器	Bluetooth#scan() もしくは Bluetooth#searchAndSelectDevice()、Bluetooth#startPairing() 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
起動可能外部機器	StarApplicationManager#addRemoteLauncher() で登録された起動可能外部機器(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 SPP 通信 API 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。	127.0.0.1

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 6-11）。
（削除したいデバイスが起動可能外部機器としても登録されている場合は、起動可能外部機器一覧からも同時に削除されます）

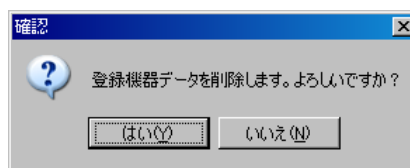


図 6-11 登録機器削除確認画面

4. 起動可能外部機器登録を削除するには、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 6-12）。

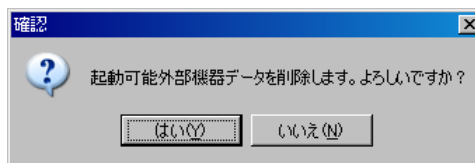


図 6-12 起動可能外部機器削除確認画面

ここに設定した自 Bluetooth デバイス情報が対向側（別のエミュレータ上）の i アプリへ「接続先相手の情報」として引き渡されます。また、対向側の i アプリで scan()、searchAndSelectDevice() を実施したときに表示される機器選択画面で表示されます（図 6-13）。



図 6-13 Bluetooth API 実行時の外部機器選択画面

エミュレータでは、Bluetooth シンプルペアリング機能による接続 API (Bluetooth#startPairingByOOB()) を使用した場合も、これと同様の外部機器選択画面が表示されます (Bluetooth#startPairingByOOB() を使用した場合は、対向側でも Bluetooth#startPairingByOOB() を使用している機器のみが表示されます)。

6.6.2. Bluetooth HDP 設定

Bluetooth HDP 機器に関する設定を行います。

Bluetooth HDP 機器の設定をするには：

1. メニューから「設定」→「アプリケーション動作環境設定...」→「Bluetooth HDP 設定」タブを選択します (図 6-14)。

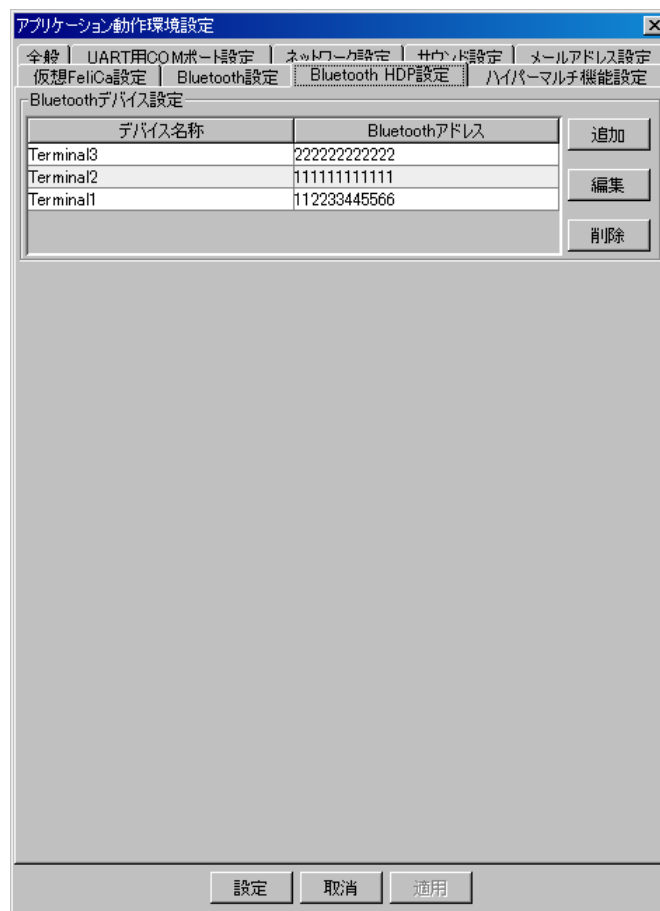


図 6-14 Bluetooth HDP 設定画面

2. Bluetooth HDP機器を追加したい場合は「追加」ボタンを押します。またHDP機器の設定を変更したい場合は、一覧からHDPデバイスを選択して「編集」ボタンを押します。HDP設定画面が表示されます（図 6-15）。

図 6-15 HDP 設定画面

3. 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 6-5 に示すとおりです。

表 6-5 HDP 設定項目

設定項目	説明	デフォルト
デバイス名	Bluetooth デバイス名称を設定します。	なし
Bluetooth アドレス	Bluetooth アドレスを設定します。	なし
端末属性情報	端末属性を選択します。	携帯電話
Spec Data	Extended Inquiry Response に含まれている Manufacturer Specific Data を設定します。	なし
Local Name	Extended Inquiry Response に含まれている Shortened Local Name を選択します。	なし
UUID	Extended Inquiry Response に含まれている 16-bit Service Class UUIDs を選択します。（エミュレータでは 1 件のみ登録できます）	なし
MDEP 登録一覧	HDP 機器が持つ MDEP（MCAP Data End Point）の一覧が表示されます。	なし

4. MDEP（MCAP Data End Point）の情報を追加したい場合は「追加」を押します。また MDEP情報の設定を変更したい場合は、一覧から MDEP情報を選択して「編集」を押します。
MDEP設定画面が表示されます（図 6-16）。



図 6-16 MDEP 設定画面

5. 各設定項目を設定し、「設定」を押します。

各設定項目は表 6-6 に示すとおりです。

表 6-6 MDEP 設定項目

設定項目	説明	デフォルト
ID	MDEP ID を設定します。 (0~255 の整数値)	なし
Type	MDEP Data Type を選択します。 (血圧計／体温計)	血圧計
Role	MDEP Role を選択します。 (Source／Sink)	Source

6. HDP 設定画面で「設定」を押します。
Bluetooth HDP 設定画面に追加・変更内容が反映されます。
(MDEP が 1 件も登録されていない状態では「設定」操作は行えません)
7. Bluetooth HDP機器を削除するには、削除したいHDP機器を選択して「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 6-17)。

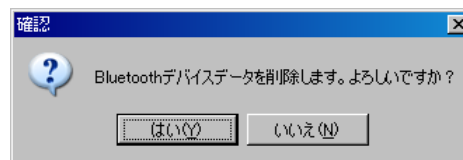


図 6-17 HDP 削除確認画面

選択したHDP機器のBluetoothアドレスが登録機器一覧に含まれている場合、HDP設定の削除は行えません。Bluetooth設定画面（「6.6.1 Bluetooth設定」参照）で登録機器一覧からの削除を行ってください。

8. 「設定」もしくは「適用」を押します。

6.7. 仮想 FeliCa 設定

「iα ppliTool」の仮想 FeliCa の動作を設定することにより、FeliCa 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通信を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 6-18）。

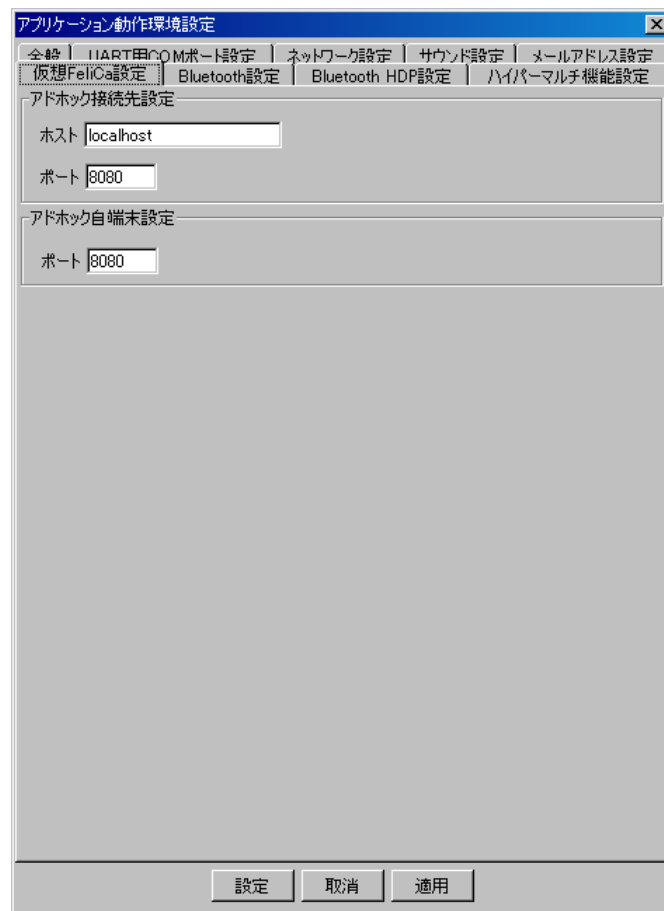


図 6-18 仮想 FeliCa 設定画面

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 6-7 に示すとおりです。

表 6-7 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	LAN 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	LAN 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

※ FeliCaアドホック通信では、クライアント側で実行するAdhocDataTransfer#setup()の第 1 引数に設定するADFのURL（絶対パス）は、サーバ側のADFのURLと一致する必要があります。従って、サーバ側の i アプリではアプリケーション動作環境設定画面の「ネットワーク設定」タブで「ADFのURL」を設定し（「6.1 ネットワークの設定」参照）、クライアント側の i アプリではAdhocDataTransfer#setup()の第 1 引数として上記で設定したADFのURLを指定します。

6.8. ハイパーマルチ機能設定

「iα ppliTool」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 アプリからの携帯電話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携帯電話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 6-19）。



図 6-19 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 6-8 に示します。

表 6-8 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。

第7章 ネイティブデータ編集

この章では、i アプリにて操作可能な個々の機能の設定について説明します。
この章で説明する内容は、メニューから「設定」→「ネイティブデータ編集...」を選択したときの各種機能です。

7.1. 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 7-1）。

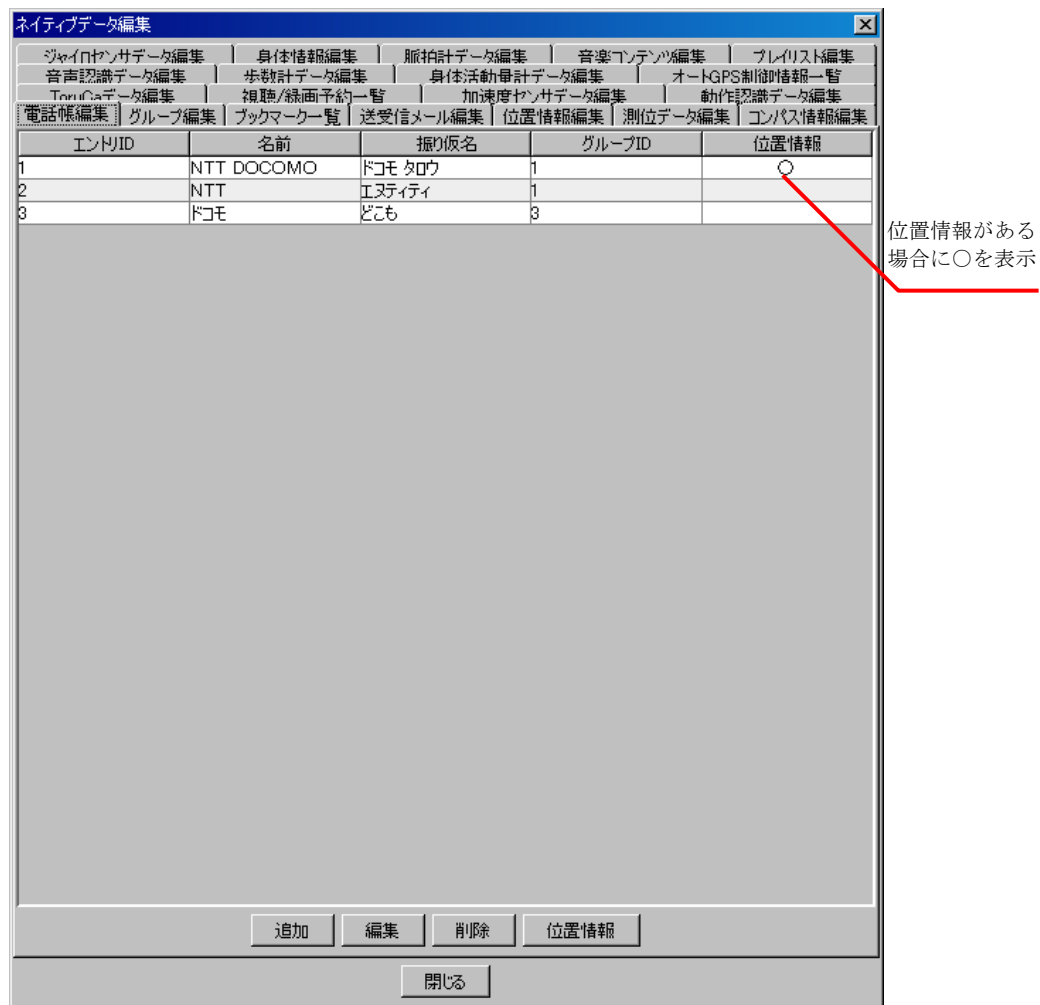


図 7-1 電話帳エントリ一覧画面

2. 電話帳エントリ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を選択して、「編集」ボタンを押します（図 7-2）。

図 7-2 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 7-1 に示すとおりです。

表 7-1 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 7-5）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 7-3）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]

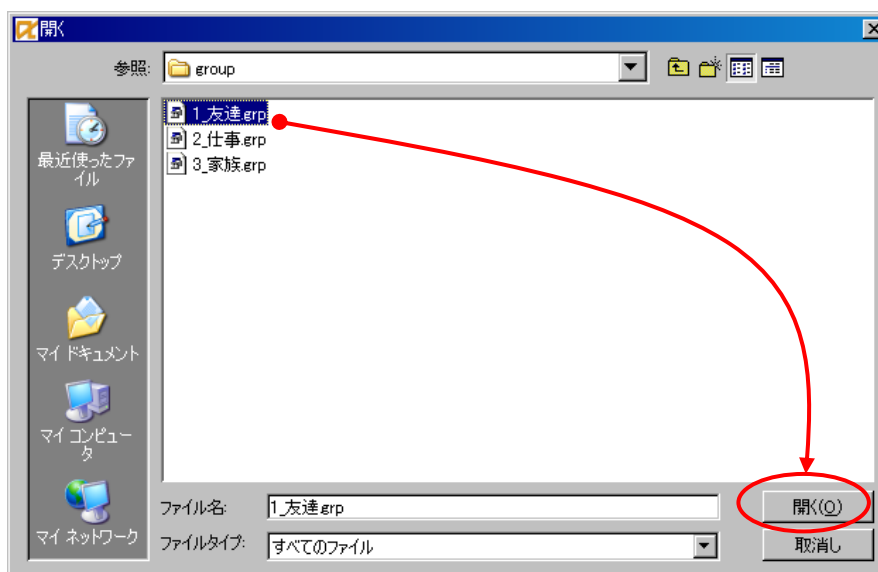


図 7-3 グループの選択ダイアログ

ここで作成した電話帳エントリは、
C:¥iDKStar2.0¥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（「7.5. 位置情報編集」参照）。



図 7-4 位置情報登録

表 7-2 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	GPS 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)	なし
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。	0
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「○」に変わります(図 7-1)。
- 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(「4.21. 位置情報電話帳起動」参照)。

7.2. 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 7-5）。

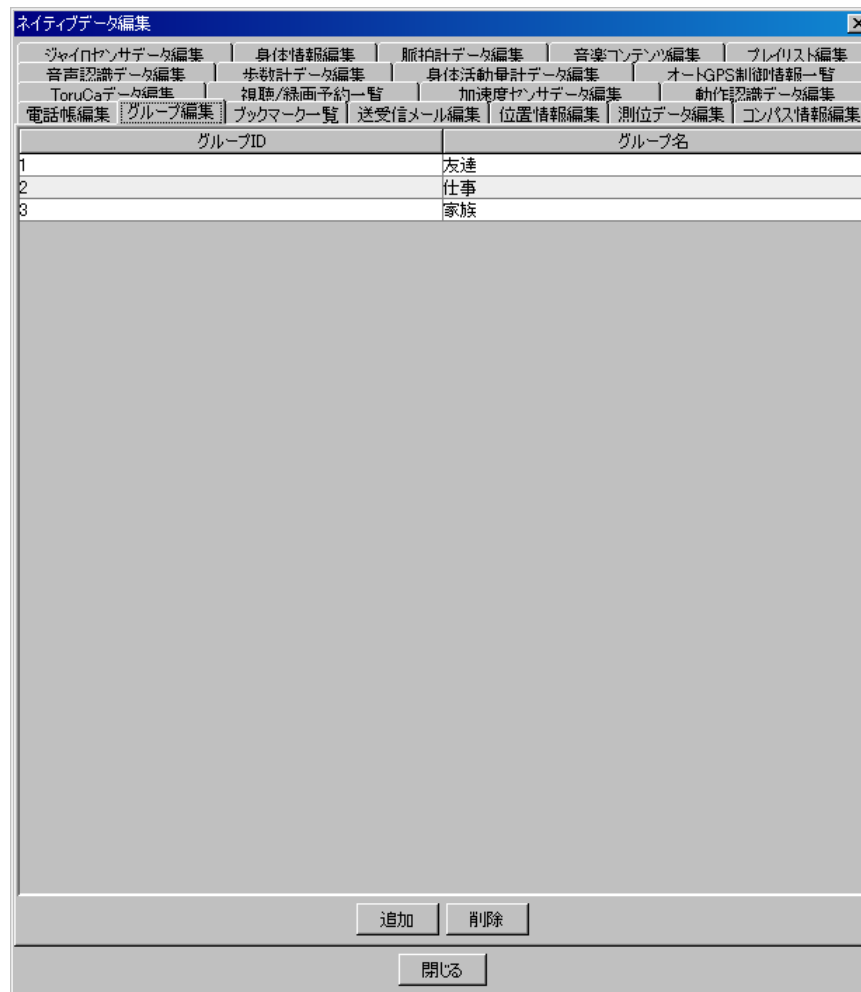


図 7-5 グループ一覧画面

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 7-6）。

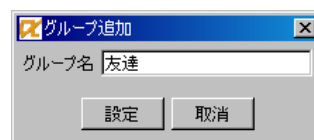


図 7-6 グループ追加画面

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、

C:\¥iDKStar2.0¥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。

7.3. 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 7-7）。

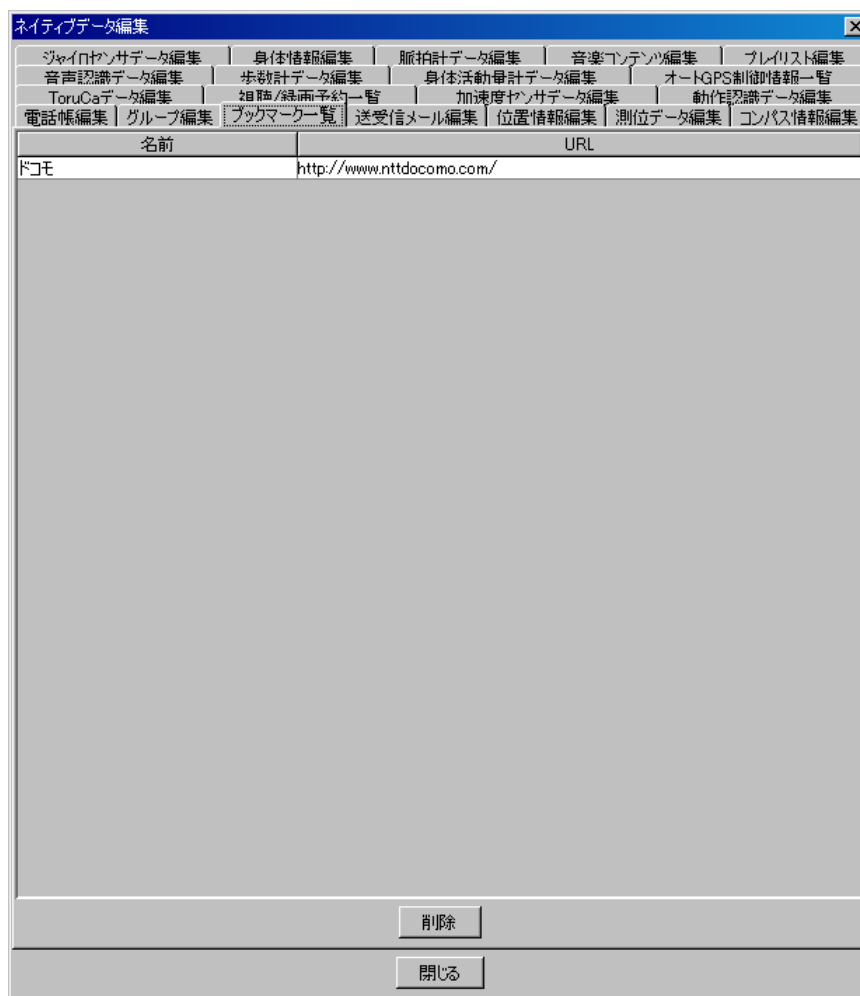


図 7-7 ブックマーク一覧

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。

7.4. 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 7-8）。



図 7-8 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 7-8）。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 7-9、図 7-10）。

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 : 2009 年 07 月 17 日 15 時 58 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 7-9 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 : 2009 年 07 月 17 日 16 時 01 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 7-10 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 7-3 に示すとおりです。

表 7-3 送受信メールの設定項目

設定項目	説明	デフォルト
To	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
From	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	ADF に設定した MessageCode。	なし
ネイティブメール	ネイティブメール（携帯電話が管理するメール＝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（API 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 7-11）。

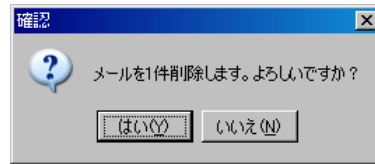


図 7-11 メール削除確認画面

メールファイルの形式については、「15.6. 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。
（「4.8. 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。

7.4.1. SMS の扱いについて

SMS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 7-12、図 7-13）。

- 受信ボックス
From : 電話番号
Subject : SMS
ネイティブメール
- 送信ボックス
To : 電話番号
Subject : SMS
ネイティブメール
- 未送信ボックス
To : 電話番号
Subject : SMS
ネイティブメール

メール編集(受信ボックス)

To : message_app_usr0@docomo.ne.jp.

From : 090xxxxxxx

タイトル : SMS

受信日 : 2009 年 07 月 17 日 16 時 04 分

MessageCode : 0000000000

ネイティブメール : ☐

状態 : ☒ 既読 ☐ 未読

本文

おはようございます

設定 取消

図 7-12 受信ボックスの SMS 作成

メール編集(送信ボックス)

To : 090xxxxxxxx

From : message_app_usr0@docomo.ne.jp

タイトル : SMS

送信日 : 2009 年 07 月 17 日 16 時 05 分

MessageCode : 0000000000

ネイティブメール : ☐

宛先状態 : ☒ XString ☐ 通常文字列

本文

こんにちは

設定 取消

図 7-13 送信ボックス／未送信ボックスの SMS 作成

7.4.2. 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（「4.9. トラストド動作の設定」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。
（「4.8. 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 7-14）。

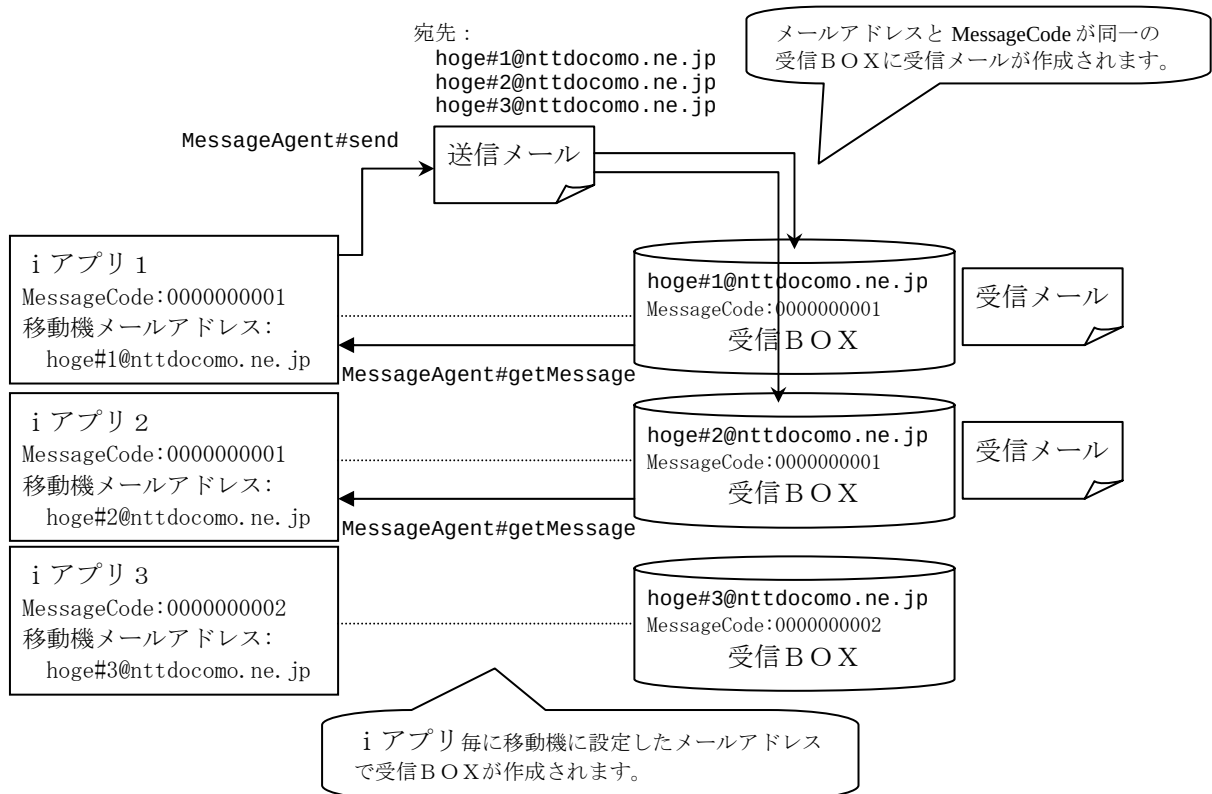


図 7-14 i アプリによるメールの送受信

※ 同一 PC 上でエミュレータを複数起動することはできません。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。

7.5. 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 7-15）。

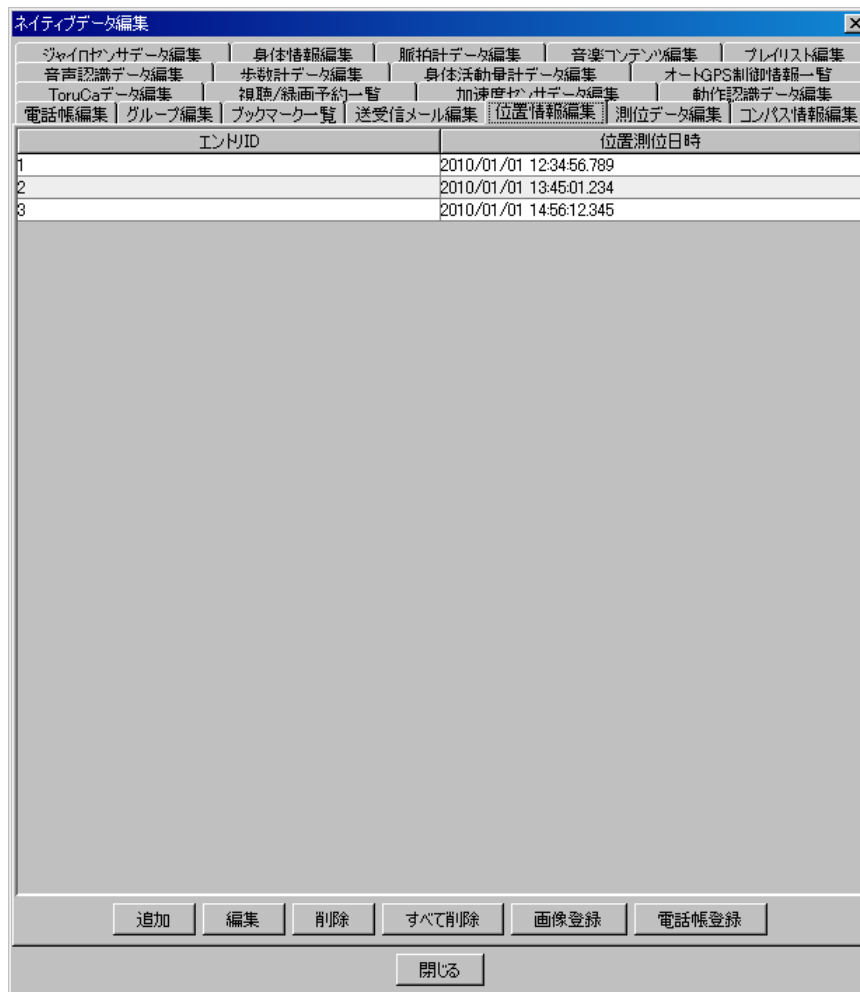


図 7-15 位置情報一覧画面

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 7-16）。
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 7-16 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 7-17）。

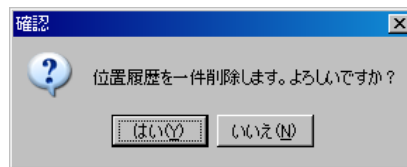
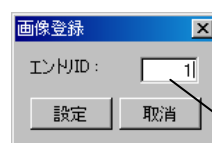


図 7-17 位置情報削除確認画面

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 7-18）。
ここで設定したエン트리IDで 位置情報画像起動ができます（「4.20. 位置情報画像起動」参照）。



前回設定したエン트리 ID

図 7-18 位置情報画像エン트리 ID 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트리IDの設定画面が表示されるので、電話帳のエン트리IDを指定して「設定」を押します（図 7-19）。
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（「7.1. 電話帳の編集」、「4.21. 位置情報電話帳起動」参照）。

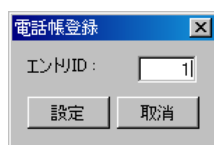


図 7-19 位置情報電話帳エントリ ID 設定画面

7.6. 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 7-20）。

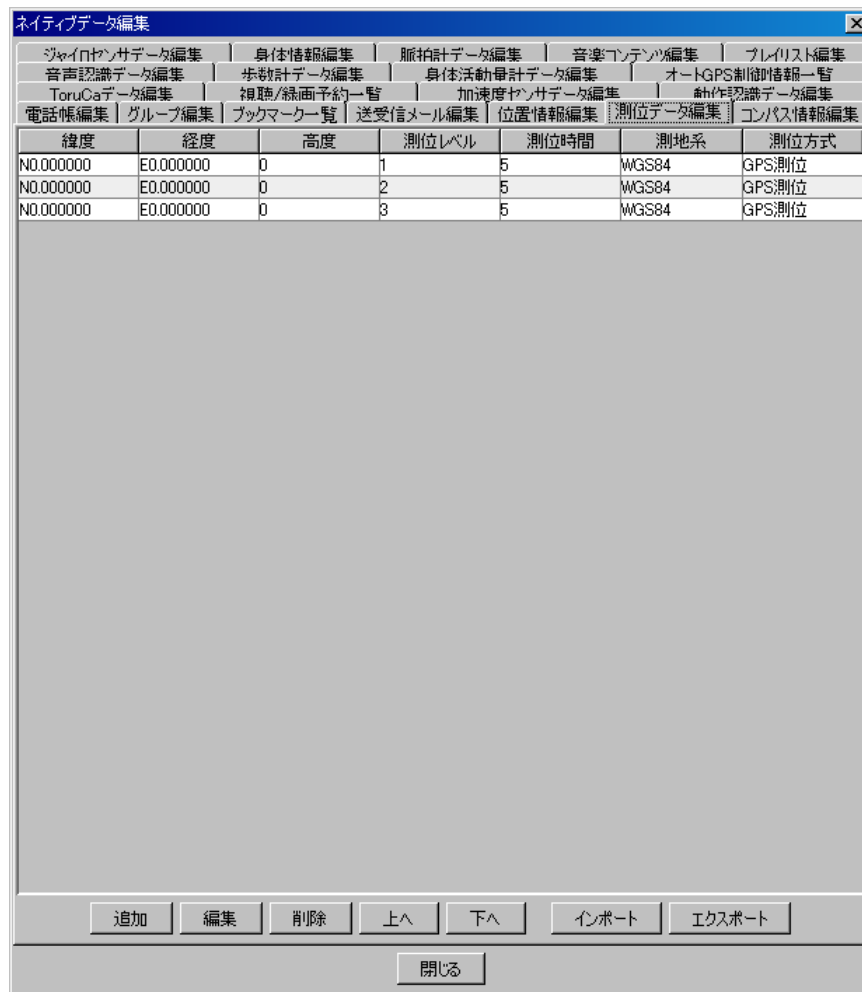


図 7-20 測位データ一覧画面

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 7-21）。



図 7-21 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 7-4 に示すとおりです。

表 7-4 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 7-22）。

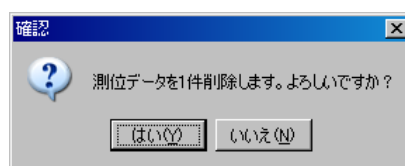


図 7-22 測位データ削除確認画面

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」を押すと、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」を押すと、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。

7.6.1. 測位データのインポート・エクスポート

【インポート機能について】

1. 「インポート」を押すと、ファイル名を入力する画面が表示されます（図 7-23）。

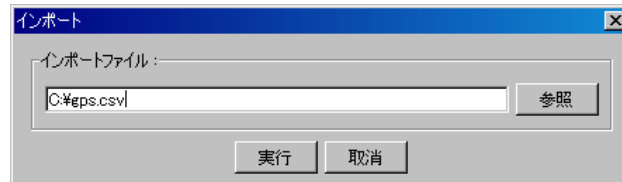


図 7-23 インポート画面

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」を押してファイルを指定し「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 7-24）、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。

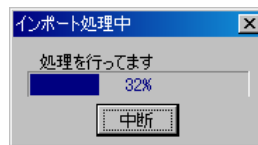


図 7-24 インポート処理中画面

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 7-25）。
「中断」を押した場合、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。

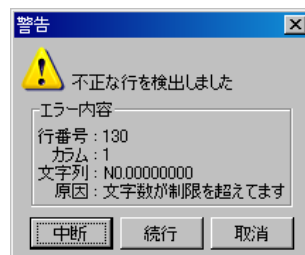


図 7-25 入力データ警告画面

【エクスポート機能について】

1. 「エクスポート」を押すと、ファイル名を入力する画面が表示されます（図 7-26）。

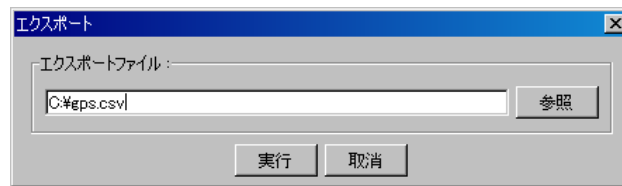


図 7-26 エクスポート画面

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」を押してファイルを指定し「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 7-27）、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。

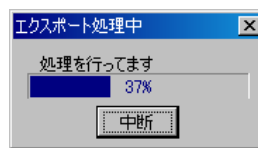


図 7-27 エクスポート処理中画面

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 7-28）。
「上書き」を押した場合、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」を押した場合、処理を終了します。

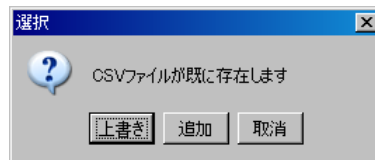


図 7-28 エクスポート処理選択画面

7.7. 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 7-29）。

磁北から直面Y軸方向への方位角	磁北から直面X軸方向への方位角
29.0	119.0
28.1	118.1
355.0	85.0
0.0	90.0
0.0	90.0

図 7-29 コンパス情報編集画面

2. 電子コンパスデータを追加したい場合は、「追加」ボタンを押します。
電子コンパスデータを編集したい場合は、一覧から編集したい電子コンパスデータを選択して「編集」ボタンを押します（図 7-30）（一覧をダブルクリックしても同じように操作できます）。

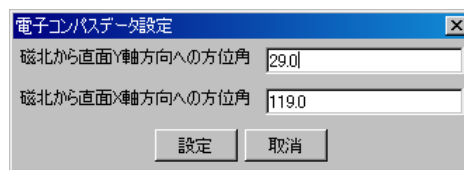


図 7-30 電子コンパスデータ設定画面

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 7-5 に示すとおりです。

表 7-5 電子コンパスデータ情報の設定項目

設定項目	説明	デフォルト
磁北から直面 Y 軸方向の方位角	直面 Y 軸の正（上）方向の磁北からの方位角を実数値（DEGREE）で入力します。（360 未満）	なし
磁北から直面 X 軸方向の方位角	直面 X 軸の正（右）方向の磁北からの方位角を実数値（DEGREE）で入力します。（360 未満）	なし

4. 電子コンパスデータを削除したい場合は、一覧から削除したい電子コンパスデータを選択して「削除」ボタンを押します。
削除しても良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 7-31）。

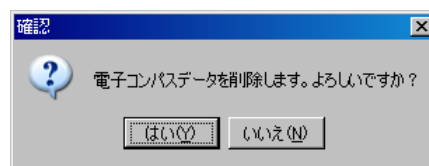


図 7-31 電子コンパスデータ削除確認画面

7.8. 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 7-32）。

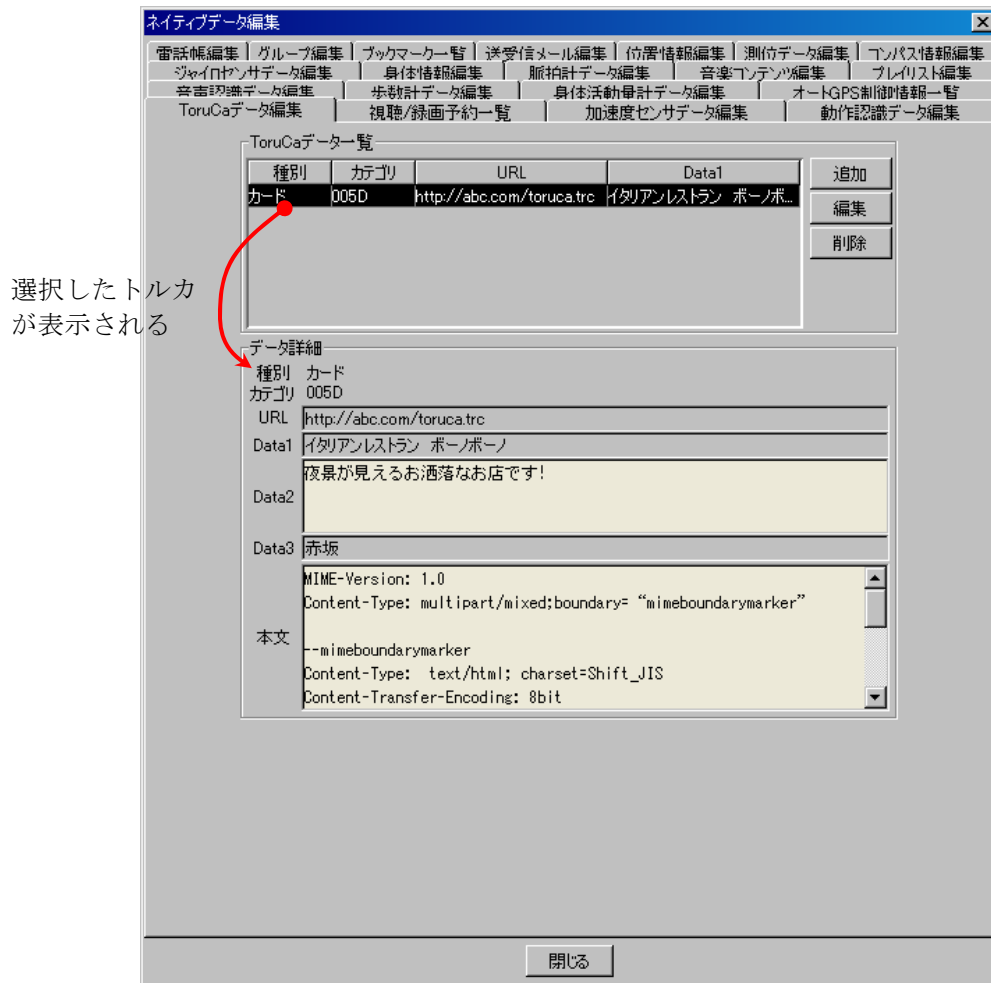


図 7-32 トルカデータ一覧画面

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 7-33）。
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: ファッション/アクセサリ/雑貨/インテリア

コード: 005D

バージョン: 2.0

IP-ID: 12345678

カナID: ボー

再配布識別子: 4

色ID: 9

有効期限: 061225

URL: http://abc.com/toruca.trc

Data1: イタリアンレストラン ボーノボーノ
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/mixed;boundary="mi
--mimeboundarymarker
Content-Type: text/html; charset=Shift_JIS
Content-Transfer-Encoding: 8bit
(本文内容)

設定 取消

図 7-33 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 7-6 に示すとおりです。

表 7-6 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。 実際の携帯電話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 ID 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 ID 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]
URL	URL を指定します。 URL は実際の携帯電話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]

7.9. 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 7-34）。

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

電話帳編集	グループ編集	ブックマーク一覧	送受信メール編集	位置情報編集	測位データ編集	コンパス情報編集
マイコンセンサデータ編集	身体情報編集	脈拍計データ編集	音楽コンテナ編集	プレイリスト編集		
音声認識データ編集	歩数計データ編集	身体活動量計データ編集	オートGPS制御情報一覧			
ToruCaデータ編集	視聴/録画予約一覧	加速度センサデータ編集	動作認識データ編集			

視/録	開始時刻△	終了時刻	チャンネル名	番組名
録画	2010/01/03(Sun) 07:30	2010/01/03(Sun) 07:54	DCM	携帯野球ドコモンジャー

詳細情報

視聴/録画	録画	サービス識別	1
開始時刻	2010/01/03(Sun) 07:30	物理チャンネル	13
終了時刻	2010/01/03(Sun) 07:54	系列識別	1
チャンネル名	DCM	繰り返し	1回限り
番組名	携帯野球ドコモンジャー		

削除

閉じる

図 7-34 視聴/録画予約一覧

※ 繰り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、繰り返し種別に該当する日時のうち、「iアプリ Tool」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（図 7-35）。

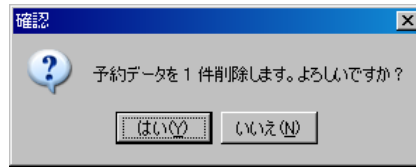


図 7-35 視聴/録画予約データ削除確認画面

7.10. 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 7-36）。

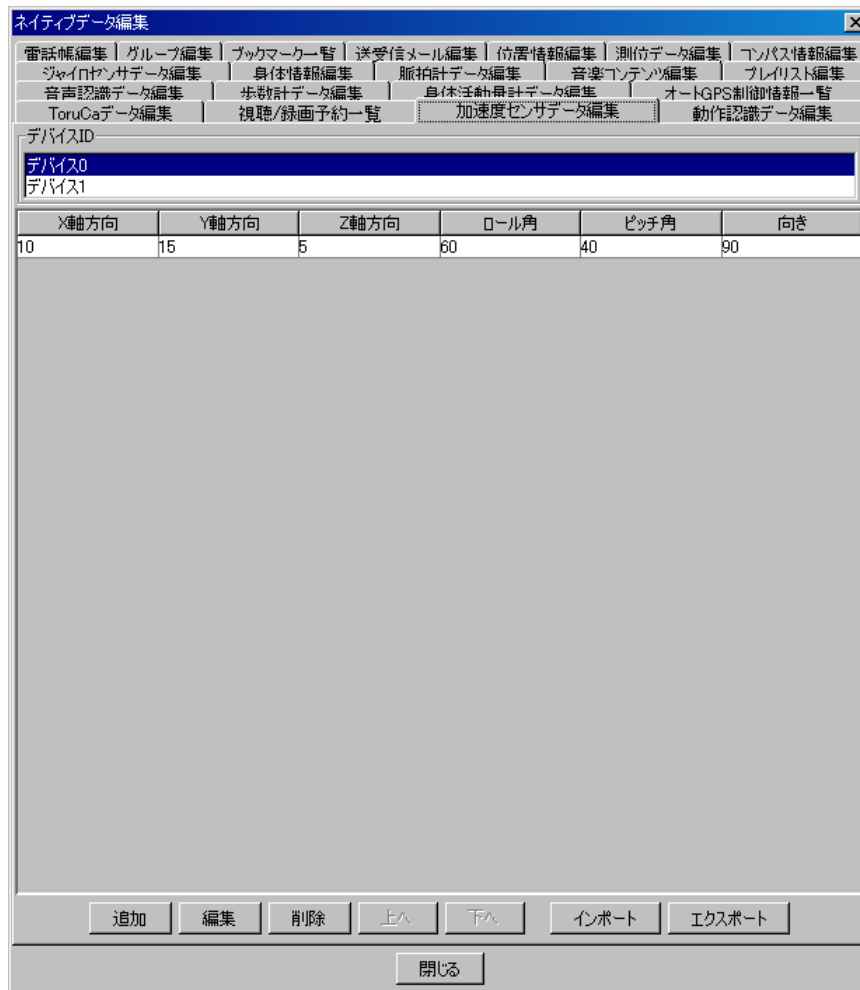


図 7-36 加速度センサデータ編集

2. 編集するデバイス ID を選択します。
加速度センサデータは、デバイス ID 毎に管理されます。デバイス ID を選択することで、選択されたデバイス ID に対する加速度センサデータが表示されます。
3. 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 7-37）。（一覧をダブルクリックしても同じように操作できます。）

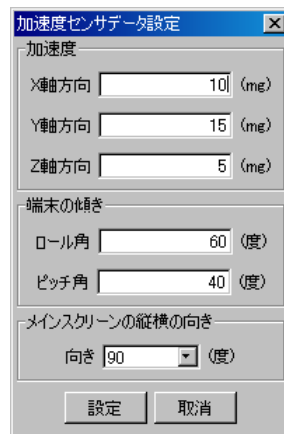


図 7-37 加速度センサデータ編集画面

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 7-7 に示すとおりです。

表 7-7 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

5. 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 7-38）。

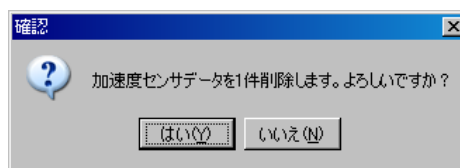


図 7-38 加速度センサデータ削除確認画面

6. 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。
7. 「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」を押すと、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。

8. 「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」を押すと、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、「7.6. 測位データ編集」を参照してください。

7.11. 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 7-39）。

ネイティブデータ編集

電話帳編集 | グループ編集 | ブックマーク一覧 | 送受信メール編集 | 位置情報編集 | 測位データ編集 | トンパス情報編集
 ジャイロセンサーデータ編集 | 身体情報編集 | 脈拍計データ編集 | 音楽コンテンツ編集 | プレイリスト編集
 音声認識データ編集 | 歩数計データ編集 | 身体活動量計データ編集 | オートGPS制御情報一覧
 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**

☒ Roll動作 ☐ Shake動作

Roll動作

X方向位置	Y方向位置	品質
100	80	RELIABLE
50	0	RELIABLE

Shake動作

振動量

0

追加 編集 削除 上へ 下へ インポート エクスポート

閉じる

図 7-39 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 7-40）。

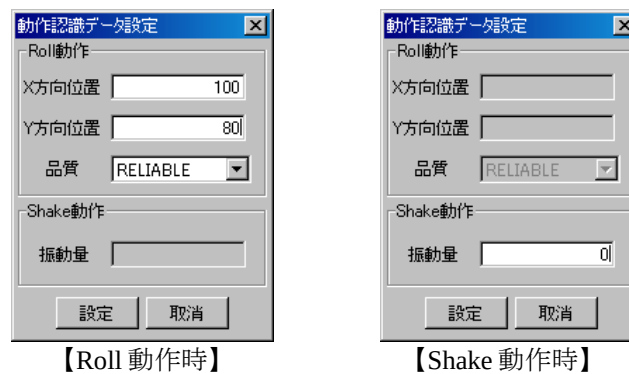


図 7-40 動作認識データ編集画面

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 7-8 に示すとおりです。

表 7-8 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 7-38）。

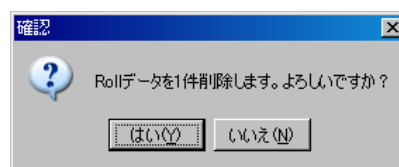


図 7-41 動作認識データ削除確認画面

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」を押すと、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」を押すと、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、「7.6.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携帯電話の移動量（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入力しております。

- 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、「4.6. 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帰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。Star 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。

7.12. 音声認識データ編集

i アプリから利用する音声認識機能の認識結果データを編集します。
エミュレータではバックエンドサーバに対し実際に DDP 通信は起こりません。本設定をおこなうことにより、指定したとおりに DDP 通信のレスポンスをエミュレートできます。また、特徴量データは一定間隔(デフォルトは 240msec)毎にダミー値を返します。(ダミー値は音声取得情報設定ファイルの値を直接変更することにより調節可能です)

音声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...」→「音声認識データ編集」タブを選択します（図 7-42）。

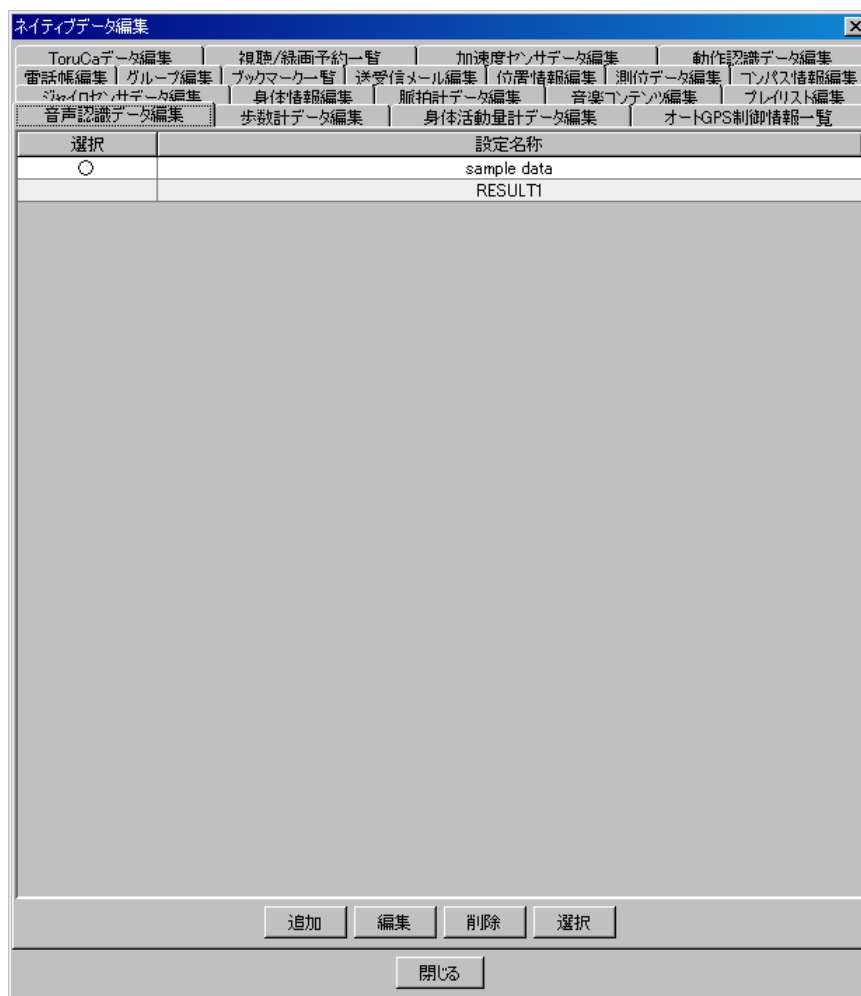


図 7-42 音声認識データ編集

2. API で使う認識結果データを選択する場合は、認識結果データを一覧から選択して「選択」ボタンを押します。選択欄が”○”となっている設定が有効となります。
3. 認識結果データを削除したい場合は、一覧から削除した認識結果データを選択して「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 7-43)。
ただし、選択状態(選択欄が ○)の認識結果データは削除できません。

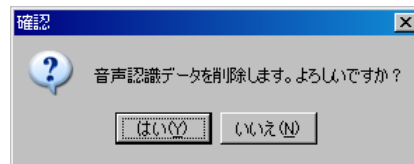


図 7-43 音声認識データ削除確認画面

4. 認識結果データを追加したい場合は、「追加」ボタンを押します。
認識結果データを編集したい場合は、一覧から編集したい認識結果データを選択して「編集」ボタンを押します(図 7-44)。(一覧をダブルクリックしても同じように操作できます。)

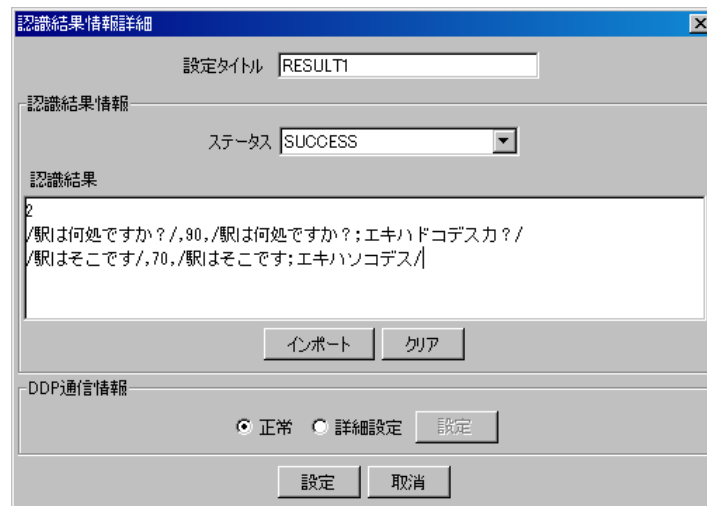


図 7-44 認識結果情報詳細画面

5. 各設定項目を設定して「設定」を押します。

各設定項目は表 7-9 に示すとおりです。

表 7-9 認識結果情報の設定項目

設定項目	説明	デフォルト
設定タイトル	名称を指定します。	なし
ステータス	認識結果のステータスコードを選択します。	0
認識結果	音声認識バックエンドサーバから取得された結果チャネル(ResultType)を指定します。	なし
DDP 通信情報	DDP 通信における通信状況の設定をします。 「正常」を選択した場合は、全ての DDP コマンドが正常終了となります。	正常

6. 「インポート」ボタンを押すと、認識結果をファイルから読み込みます。

7. 「クリア」ボタンを押すと、認識結果の入力をクリアします。
8. DDP通信情報の「設定」を押すと、DDP通信を実行した結果の詳細設定画面を表示します（図 7-45）。
もし、DDP通信の振る舞いを設定したい場合は、各種コマンド発行時のエラー情報（サーバレスポンスコード）を指定してください。

DDP通信情報編集

ネゴシエーション
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

送信
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG
送信回数 ☐ 指定なし ☒ 指定あり 1 回

結果取得
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

キャンセル
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

終了
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

設定 取消

図 7-45 DDP 通信情報編集画面

9. 各設定項目を設定して「設定」を押します。
各設定項目は 表 7-10 に示すとおりです。

表 7-10 DDP 通信情報の設定項目

設定項目	説明	デフォルト
ネゴシエーション	擬似 DDP 通信にてリクエストヘッダに NEGOTIATE コマンドが含まれていた場合のレスポンスコードを指定します。	正常 200
送信	擬似 DDP 通信にてリクエストヘッダに RECOGNIZE コマンドが含まれていた場合のレスポンスコードを指定します。	正常 200
送信回数	擬似 DDP 通信にてリクエストヘッダに RECOGNIZE コマンドが含まれていた場合、何回目のレスポンスでステータス情報を” 200 OK”にするか？の回数を指定します。指定なしの場合は 1 回目となります。指定ありの場合は（指定された回数-1）回目までは” 100 Continue”を返します。	指定なし
結果取得	擬似 DDP 通信にてリクエストヘッダに GET-RESULT コマンドが含まれていた場合のレスポンスコードを指定します。	正常 200
キャンセル	擬似 DDP 通信にてリクエストヘッダに CANCEL コマンドが含まれていた場合のレスポンスコードを指定します。	正常 200
終了	擬似 DDP 通信にてリクエストヘッダに CLOSE コマンドが含まれていた場合のレスポンスコードを指定します。	正常 200

7.13. 歩数計データ編集

i アプリから利用する歩数計機能の結果データを編集します。

歩数計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...」→[歩数計データ編集] タブを選択します（図 7-46）。

ネイティブデータ編集

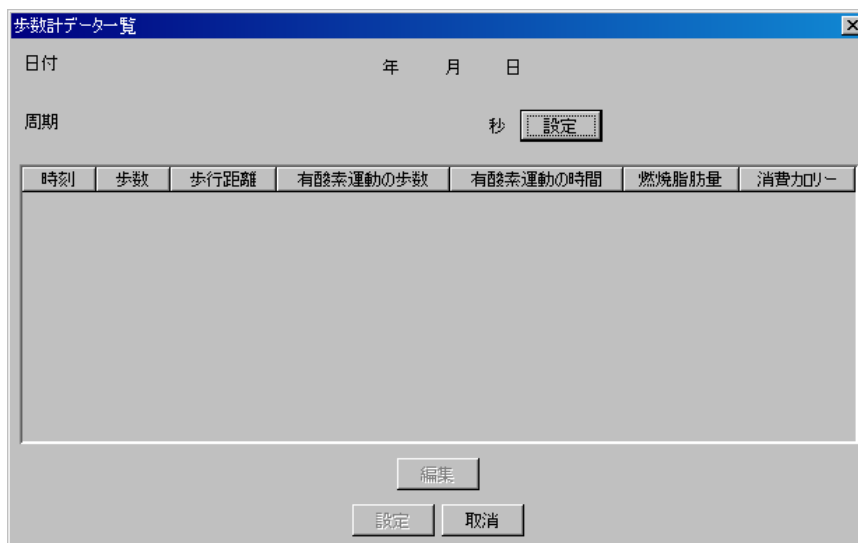
日付	歩数	歩行距離	燃焼脂肪量	消費カロリー
2010/08/01	6000	3500	8000	3800
2010/08/02	5500	3200	720	3000
2010/08/03	5600	3300	770	3400

追加 編集 削除

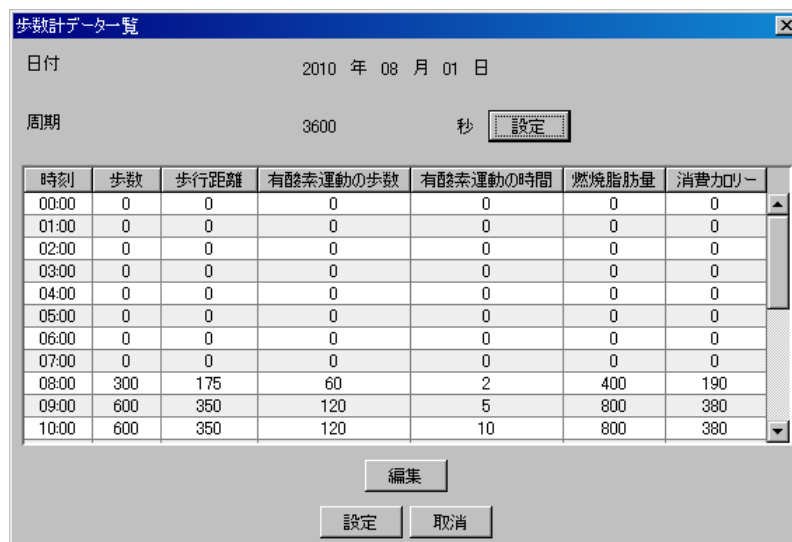
閉じる

図 7-46 歩数計データ編集

2. 歩数計データを追加したい場合は、「追加」ボタンを押します。
歩数計データを編集したい場合は、一覧から編集したい歩数計データを選択して「編集」ボタンを押します（図 7-47）。（一覧をダブルクリックしても同じように操作できます。）



【追加時】



時刻	歩数	歩行距離	有酸素運動の歩数	有酸素運動の時間	燃焼脂肪量	消費カロリー
00:00	0	0	0	0	0	0
01:00	0	0	0	0	0	0
02:00	0	0	0	0	0	0
03:00	0	0	0	0	0	0
04:00	0	0	0	0	0	0
05:00	0	0	0	0	0	0
06:00	0	0	0	0	0	0
07:00	0	0	0	0	0	0
08:00	300	175	60	2	400	190
09:00	600	350	120	5	800	380
10:00	600	350	120	10	800	380

【編集時】

図 7-47 歩数計データ一覧画面

3. 歩数情報データを追加したい場合は、まず周期表示の右側の「設定」ボタンを押して、日付と周期を設定します（図 7-48）。

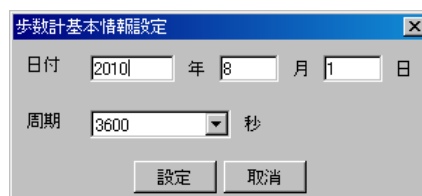


図 7-48 歩数計基本情報設定画面

4. 各設定項目を設定して「設定」を押すと、歩数計データ一覧画面に戻り、歩数計データの設定が可能になります。

各設定項目は表 7-11 に示すとおりです。

表 7-11 歩数計基本情報の設定項目

設定項目	説明	デフォルト
日付	歩数計データの日付を入力します。 すでに登録済みの日付を入力した場合は、エラーになります。	なし
周期	その日の歩数計の測定周期を選択します。 選択肢は 60 秒単位で、1 時間（3,600 秒）を丁度割り切れる秒数のみ指定可能です。	60 秒

5. 一覧から編集したい計測周期のデータを選択して「編集」ボタンを押します（図 7-49）。（一覧をダブルクリックしても同じように操作できます。）

図 7-49 歩数計データ設定画面

6. 各設定項目を設定して「設定」を押します。

各設定項目は表 7-12 に示すとおりです。

表 7-12 歩数計データ情報の設定項目

設定項目	説明	デフォルト
歩数	その計測周期の歩数を入力します。	なし
歩行距離	その計測周期の歩行距離を入力します。	なし
有酸素運動の歩数	その計測周期の有酸素運動の歩数を入力します。	なし
有酸素運動の時間	その計測周期の有酸素運動の時間を入力します。	なし
燃焼脂肪量	その計測周期の燃焼脂肪量を入力します。	なし
消費カロリー	その計測周期の消費カロリーを入力します。	なし

7. 歩数計データを設定して「設定」を押すと、歩数計データ一覧画面に戻ります。
8. 編集が完了したら、歩数計データ一覧画面で「設定」を押すと、歩数計データ編集画面に戻ります。

9. 歩数計データを削除したい場合は、一覧から削除したい歩数計データを選択して「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 7-50）。

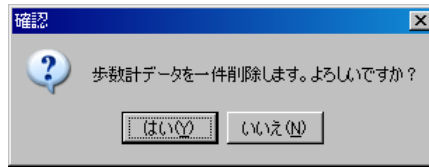


図 7-50 歩数計データ削除確認画面

7.14. 身体活動量計データ編集

i アプリから利用する身体活動量計機能の結果データを編集します。

身体活動量計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...」→「身体活動量計データ編集」タブを選択します（図 7-51）。

ネイティブデータ編集			
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 電話帳編集 グループ編集 ブックマーク一覧 送受信メール編集 位置情報編集 測位データ編集 マップ情報編集 ジャイロセンサーデータ編集 身体情報編集 睡眠データ編集 音楽データ編集 プレイリスト編集 音声認識データ編集 歩数計データ編集 身体活動量計データ編集 オートGPS制御情報一覧			
運動強度		1.5	METs
		設定	
日付	エクササイズの周期	通常エクササイズ総計	有酸素運動エクササイズ総計
2009/07/17	360	0.0	0.0
追加 編集 削除			
閉じる			

図 7-51 身体活動量計データ編集

2. 身体活動量計データを追加したい場合は、「追加」ボタンを押します（図 7-52）。身体活動量計データを編集したい場合は、一覧から編集したい身体活動量計データを選択して「編集」ボタンを押します（図 7-52）。（一覧をダブルクリックしても同じように操作できます。）

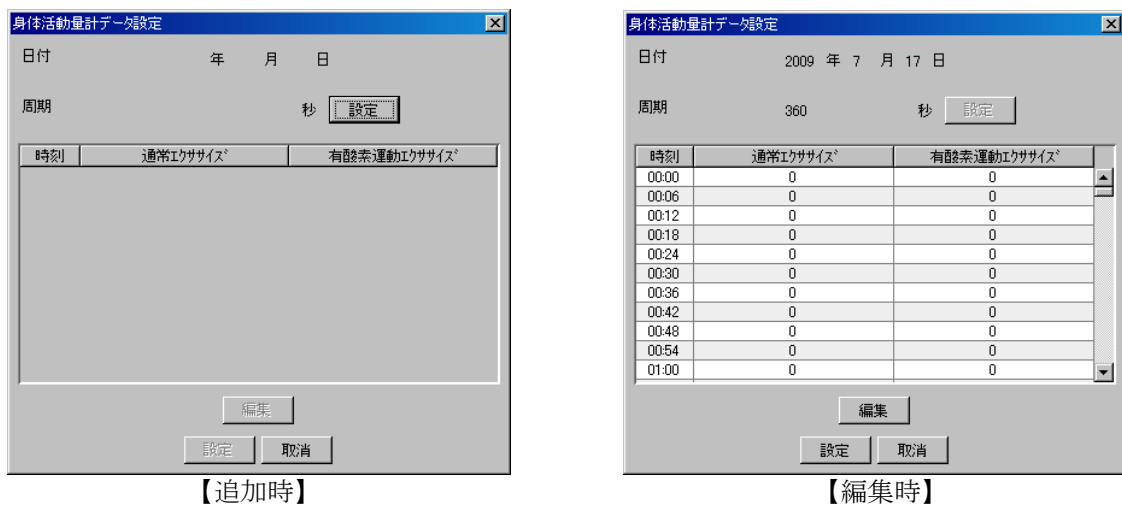


図 7-52 身体活動量計データ設定画面

- 身体活動量計データを追加したい場合は、まず周期表示の右側の「設定」ボタンを押して、日付と周期を設定します（図 7-53）。周期表示の右側の「設定」ボタンは、追加の場合以外は押せません。

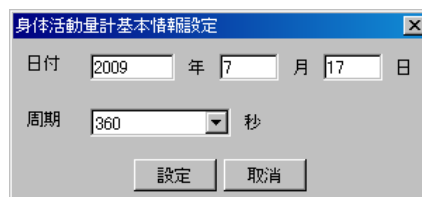


図 7-53 身体活動量計基本情報設定画面

- 各設定項目を設定して「設定」を押すと、身体活動量計データ設定画面に戻り、身体活動量計データの設定が可能になります。

各設定項目は表 7-13 に示すとおりです。

表 7-13 身体活動量計基本情報の設定項目

設定項目	説明	デフォルト
日付	身体活動量計データの日付を入力します。 すでに登録済みの日付を入力した場合は、エラーになります。	なし
周期	その日の身体活動量計の測定周期を選択します。 選択肢は 60 秒単位で、1 日（86,400 秒）を丁度割り切れる秒数のみ指定可能です。	60 秒

- 一覧から編集したい時刻のデータを選択して「編集」ボタンを押します（図 7-54）。（一覧をダブルクリックしても同じように操作できます。）
通常エクササイズ値と有酸素運動エクササイズ値をそれぞれ設定できます。
これはPhysicalActivityData#getExerciseで指定可能な抽出条件に対応しており、PhysicalActivityData.FILTER_NONE指定時は通常エクササイズ値が、

PhysicalActivityData.FILTER_AEROBICS指定時は有酸素運動エクササイズ値が取得対象になります。
エクササイズとして設定できる値は 0 以上、100 未満の範囲内（初期値は 0）です。

※

有酸素運動エクササイズ（PhysicalActivityData.FILTER_AEROBICS指定時）の値は、通常エクササイズ（PhysicalActivityData.FILTER_NONE指定時）の内訳となるため、通常エクササイズの値よりも大きい値となることはありませんが、エミュレータではこれらを個別に管理しているため、有酸素運動エクササイズの値に通常エクササイズよりも大きい値を設定した場合にも、その値がそのままFILTER_AEROBICS適用時の値としてアプリケーションプログラムに返されますのでご注意ください。

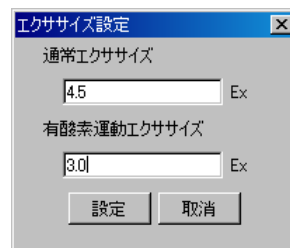


図 7-54 エクササイズ設定画面

6. エクササイズを設定して「設定」を押すと、身体活動量計データ設定画面に戻ります。
7. 編集が完了したら、身体活動量計データ設定画面で「設定」を押すと、身体活動量計データ編集画面に戻ります。
8. 運動強度を変更したい場合は、身体活動量計データ編集画面で「設定」ボタンを押します（図 7-55）。
運動強度として設定できる値は 0 以上、100 未満の範囲内（初期値は 1.5）です。

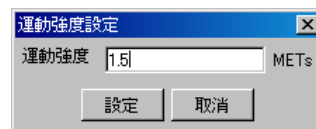


図 7-55 運動強度設定画面

9. 運動強度を設定して「設定」を押すと、身体活動量計データ編集画面に戻ります。
10. 身体活動量計データを削除したい場合は、一覧から削除したい身体活動量計データを選択して「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 7-56）。

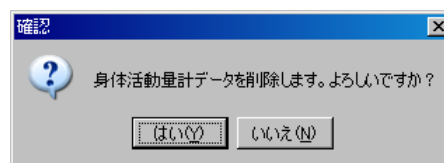


図 7-56 身体活動量計データ削除確認画面

7.15. オート GPS 制御情報一覧

i アプリから登録するオート GPS 制御情報を一覧表示、および削除します。

オート GPS 制御情報を一覧表示するには：

1. メニューから「設定」→「ネイティブデータ編集...」→「オートGPS制御情報一覧」タブを選択します（図 7-57）。

オートGPS利用アプリ名	位置情報通知先URL	位置情報通知停止期間	オートGPSフリー領域
AtfMng App	http://localhost/atfUpload/	1	

図 7-57 オート GPS 制御情報一覧

2. オートGPS制御情報を削除したい場合は、一覧から削除したいオートGPS制御情報を選択して「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 7-58）。

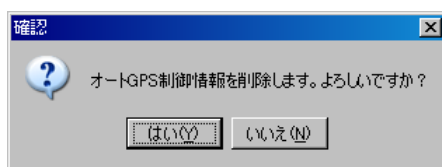


図 7-58 オート GPS 制御情報削除確認画面

7.16. ジャイロセンサデータ編集

i アプリから利用するジャイロセンサデータを編集します。

ジャイロセンサ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...」→「ジャイロセンサデータ編集」タブを選択します（図 7-59）。

ネイティブデータ編集

音声認識データ編集 | 歩数計データ編集 | 身体活動量計データ編集 | オートGPS制御情報一覧
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
電話帳編集 | ガリマ編集 | プラカマーカー一覧 | 送受信メール編集 | 位置情報編集 | 測位データ編集 | コンパス情報編集
ジャイロセンサデータ編集 | 身体情報編集 | 脈拍計データ編集 | 音楽コンテンツ編集 | プレイリスト編集

角速度範囲設定

X軸(deg/s) 最小: -180 最大: 180
Y軸(deg/s) 最小: -180 最大: 180
Z軸(deg/s) 最小: -180 最大: 180

X軸角速度値	Y軸角速度値	Z軸角速度値
50	10	-20
10	0	0
0	-20	10
-20	0	0

追加 編集 削除

閉じる

図 7-59 ジャイロセンサデータ編集

2. ジャイロ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。
ジャイロセンサデータを編集したい場合は、一覧から編集したいジャイロセンサデータを選択して「編集」ボタンを押します（図 7-60）。（一覧をダブルクリックしても同じように操作できます。）

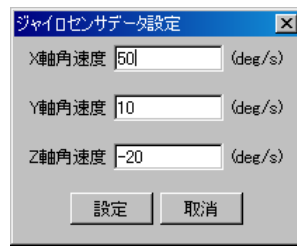


図 7-60 ジャイロセンサデータ設定画面

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 7-14 に示すとおりです。

表 7-14 ジャイロセンサデータの設定項目

設定項目	説明	デフォルト
X 軸角速度	X 軸方向(携帯電話の左から右方向)の角速度。 角速度範囲設定で指定されている範囲内の値を設定する必要があります。	なし
Y 軸角速度	Y 軸方向(携帯電話の下から上方向)の角速度。 角速度範囲設定で指定されている範囲内の値を設定する必要があります。	なし
Z 軸角速度	Z 軸方向(携帯電話の背面から前面方向)の角速度。 角速度範囲設定で指定されている範囲内の値を設定する必要があります。	なし

4. 角速度の範囲設定を変更したい場合は、ジャイロセンサデータ編集画面で「設定」ボタンを押します（図 7-61）。

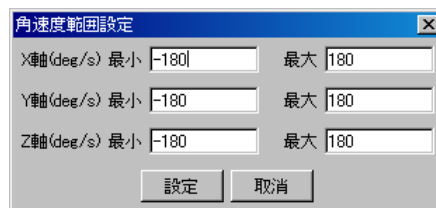


図 7-61 角速度範囲設定画面

5. 各設定項目を設定して「設定」を押します。

各設定項目は表 7-15 に示すとおりです。

表 7-15 角速度範囲の設定項目

設定項目	説明	デフォルト
X 軸最小／最大	X 軸方向の角速度として指定できる最小値／最大値。 既に設定されているジャイロセンサデータすべてを含む範囲を設定する必要があります。	-180 / 180
Y 軸最小／最大	Y 軸方向の角速度として指定できる最小値／最大値。 既に設定されているジャイロセンサデータすべてを含む範囲を設定する必要があります。	-180 / 180
Z 軸最小／最大	Z 軸方向の角速度として指定できる最小値／最大値。 既に設定されているジャイロセンサデータすべてを含む範囲を設定する必要があります。	-180 / 180

6. ジャイロセンサデータを削除したい場合は、一覧から削除したいジャイロ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 7-62）。

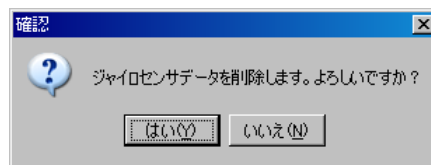


図 7-62 ジャイロセンサデータ削除確認画面

7.17. 身体情報編集

i アプリから利用する身体情報データを編集します。

身体情報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...」→「身体情報編集」タブを選択します（図 7-63）。

Native Data Edit

音声認識データ編集	歩数計データ編集	身体活動量計データ編集	オートGPS制御情報一覧
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
電話帳編集	グループ編集	Bluetoothマーク一覧	送信メール編集
位置情報編集	測位データ編集	コンパス情報編集	
ジャイロセンサーデータ編集	身体情報編集	脈拍計データ編集	音楽コンテンツ編集
			プレイリスト編集

分解能

身長:	5	(mm)
体重:	500	(g)

範囲

身長:	0	-	2100	(mm)
体重:	0	-	200000	(g)

身体情報

身長:	1800	(mm)
体重:	60000	(g)

設定

閉じる

図 7-63 身体情報編集

2. 身体情報データを追加したい場合は、「身長」「体重」の各項目に値を入力します。入力可能な範囲は、「範囲」として表示されている範囲内です。範囲外の値を入力した場合、範囲内の値に自動的に補正されます。また、「分解能」で指定されている値を単位とし、「分解能」で割り切れない値を入力した場合は、近い値に自動的に補正されます。
3. 「設定」を押します。

「範囲」および「分解能」を画面から変更することはできません。「範囲」および「分解能」を変更する場合は、身体情報データファイルを直接編集してください。（「15.48 身体情報データファイル」参照）

7.18. 脈拍計データ編集

i アプリから利用する脈拍計データを編集します。

脈拍計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...」→「脈拍計データ編集」タブを選択します（図 7-64）。

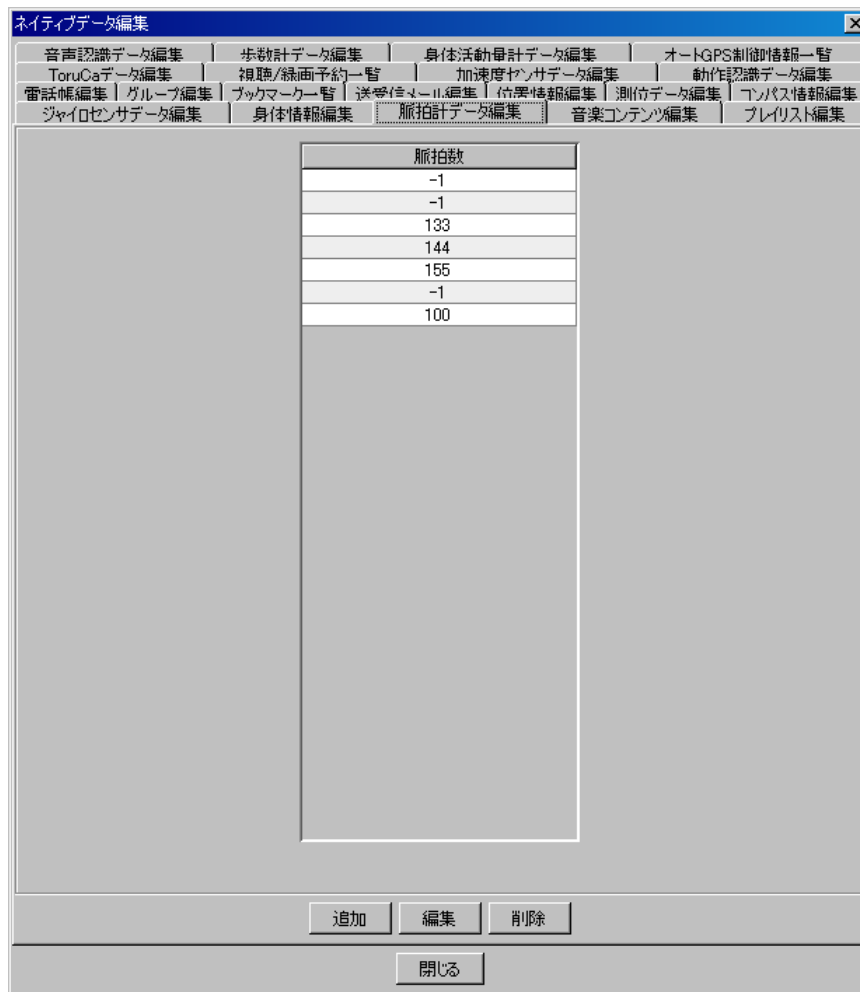


図 7-64 脈拍計データ編集

2. 脈拍計データを追加したい場合は、「追加」ボタンを押します。
脈拍計データを編集したい場合は、一覧から編集したい脈拍計データを選択して「編集」ボタンを押します（図 7-65）。（一覧をダブルクリックしても同じように操作できます。）

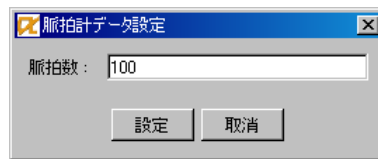


図 7-65 脈拍計データ設定画面

3. 脈拍数を設定して「設定」を押します。

各設定項目は 表 7-16 に示すとおりです。

表 7-16 脈拍計データの設定項目

設定項目	説明	デフォルト
脈拍数	脈拍数を指定します。 -1 を指定すると測定できなかった場合となります。	なし

4. 脈拍計データを削除したい場合は、一覧から削除したい脈拍計データを選択して「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 7-66）。

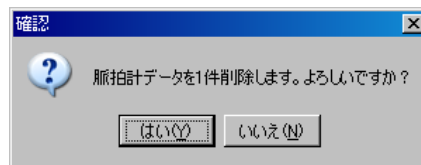


図 7-66 脈拍計データ削除確認画面

7.19. 音楽コンテンツ編集

音楽プレイヤー機能で利用する音楽コンテンツ情報を編集します。

音楽コンテンツ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...」→「音楽コンテンツ編集」タブを選択します（図 7-67）。

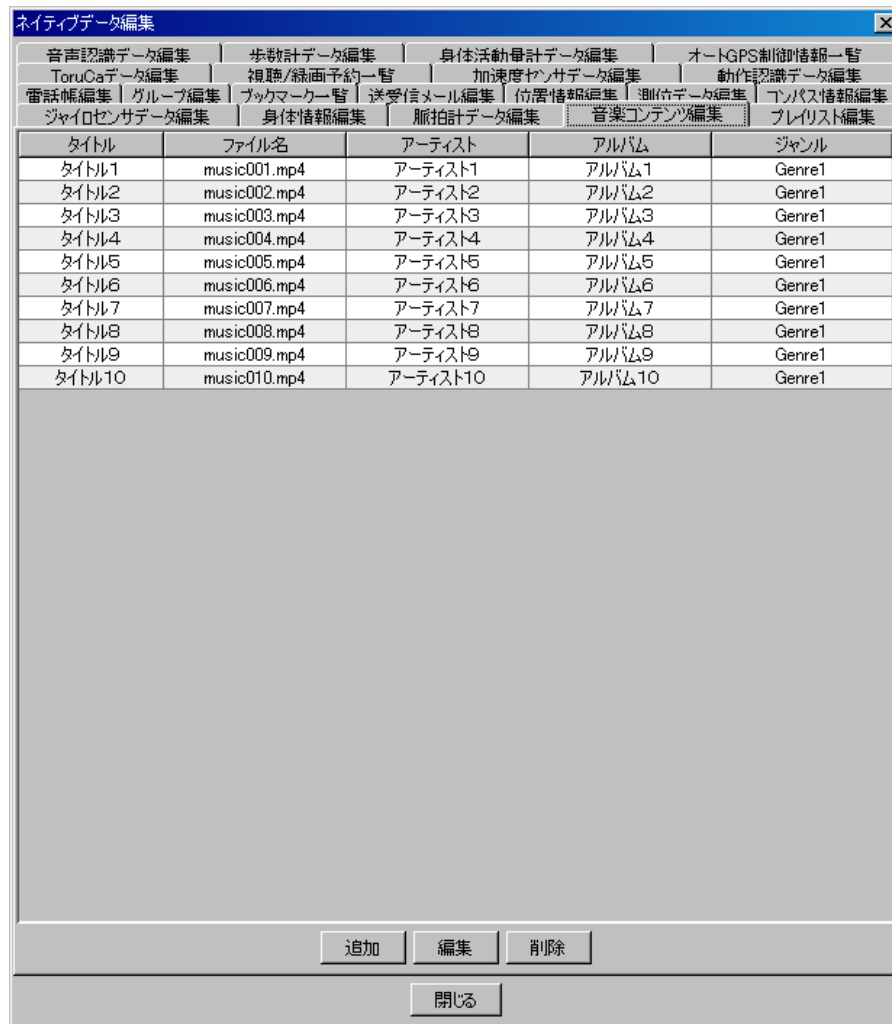


図 7-67 音楽コンテンツ情報編集画面

2. 音楽コンテンツデータを追加したい場合は、「追加」ボタンを押します。
音楽コンテンツデータを編集したい場合は、一覧から編集したい音楽コンテンツデータを選択して「編集」ボタンを押します（図 7-68）（一覧をダブルクリックしても同じように操作できます）。

図 7-68 音楽コンテンツデータ設定画面

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 7-17 に示すとおりです。

表 7-17 音楽コンテンツデータの設定項目

設定項目	説明	デフォルト
ファイル名	ファイル名を設定します。	なし
タイトル	タイトルを設定します。	なし
アーティスト	アーティストを設定します。	なし
アルバム	アルバムを設定します。	なし
ジャンル	ジャンルを設定します。	なし
ジャケット画像	ジャケット画像データのファイル名を設定します。 ジャケット画像データファイルは C:\¥iDKStar2.0¥lib¥avplayer¥image 配下に配置されている必要があります。 また、ジャケット画像を保存可能画像として扱いたい場合は「保存可能」チェックボックスをチェックします。	なし
待ち受け画像 1	1 件目の待ち受け画像データのファイル名を設定します。 待ち受け画像データファイルは C:\¥iDKStar2.0¥lib¥avplayer¥image 配下に配置されている必要があります。 また、待ち受け画像を保存可能画像として扱いたい場合は「保存可能」チェックボックスをチェックします。	なし
待ち受け画像 2	2 件目の待ち受け画像データのファイル名を設定します。	なし
再生時間	総再生時間を設定します。ミリ秒単位での設定が可能です。	1 分 0 秒

4. 音楽コンテンツデータを削除したい場合は、一覧から削除したい音楽コンテンツデータを選択して「削除」ボタンを押します。
削除しても良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 7-69）。

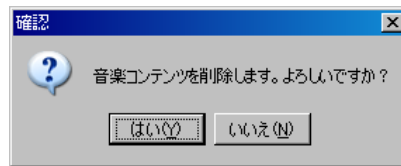


図 7-69 音楽コンテンツデータ削除確認画面

7.20. プレイリスト編集

音楽プレイヤー機能で利用するプレイリスト情報を編集します。

プレイリスト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...」→ [プレイリスト編集] タブを選択します（図 7-70）。

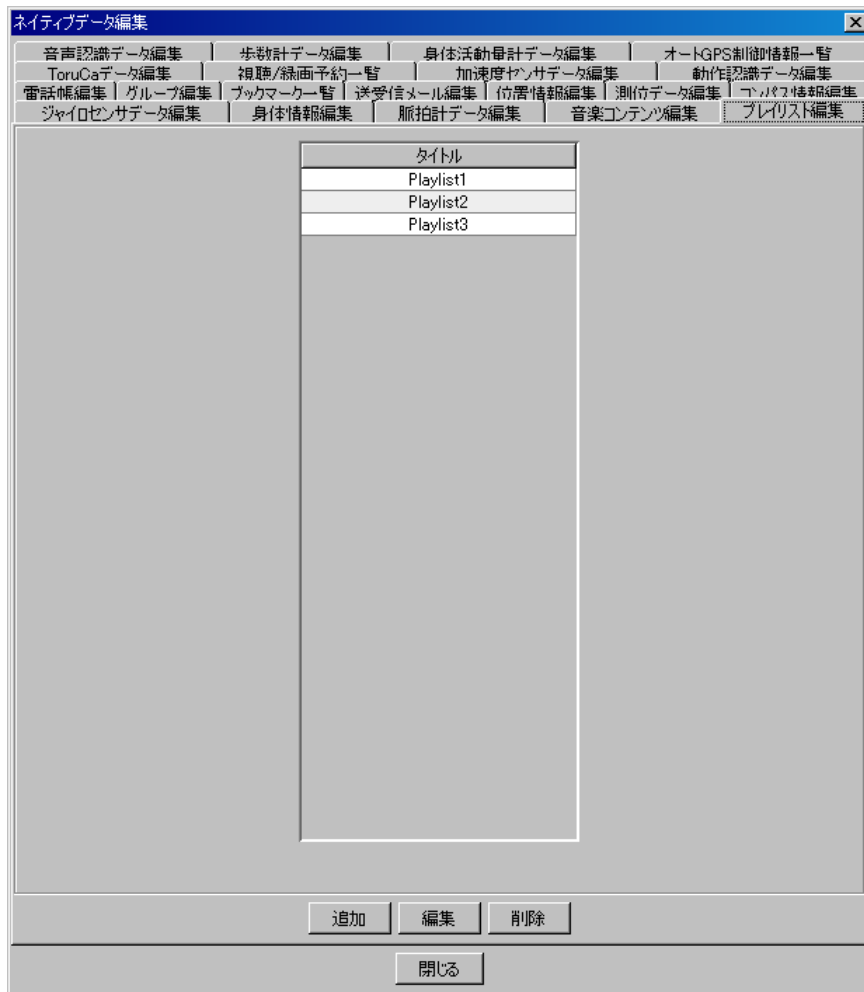


図 7-70 プレイリスト編集画面

2. プレイリストデータを追加したい場合は、「追加」ボタンを押します。
プレイリストデータを編集したい場合は、一覧から編集したいプレイリストデータを選択して「編集」ボタンを押します（図 7-71）（一覧をダブルクリックしても同じように操作できます）。

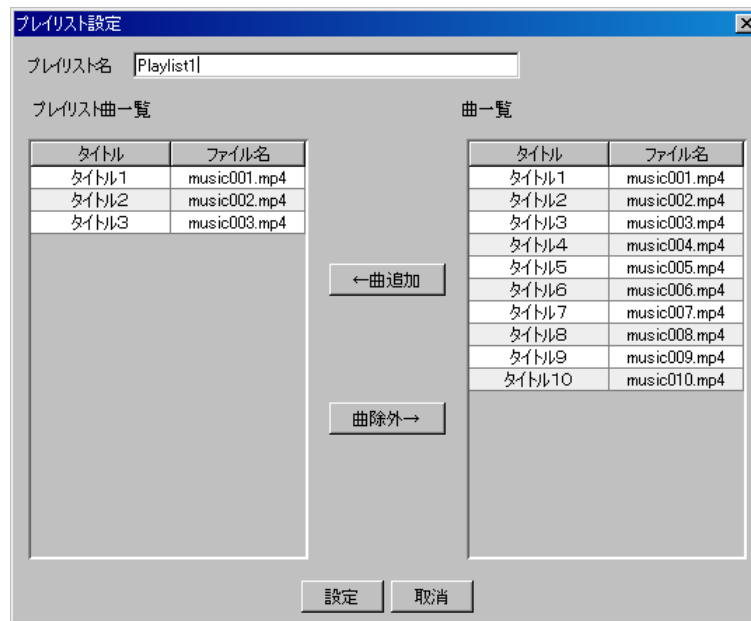


図 7-71 プレイリスト設定画面

3. プレイリスト名を設定します。（必須）
4. プレイリストに音楽コンテンツを追加する場合は、「曲一覧」から追加する音楽コンテンツを選択して「←曲追加」ボタンを押します。「曲一覧」には音楽コンテンツ編集画面で登録された音楽コンテンツが表示されます（「7.19 音楽コンテンツ編集」参照）。
選択された音楽コンテンツはプレイリスト内の一番最後に追加されます。
5. プレイリストに登録されている音楽コンテンツを削除する場合は、「プレイリスト曲一覧」から削除する音楽コンテンツを選択して「曲除外→」ボタンを押します。
6. 「設定」を押すと、設定・変更内容が反映されます。
7. プレイリストデータを削除したい場合は、一覧から削除したいプレイリストデータを選択して「削除」ボタンを押します。
削除しても良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 7-72）。

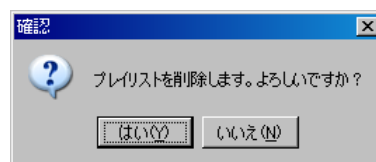


図 7-72 プレイリストデータ削除確認画面

第8章 エミュレータ環境設定

この章では、i アプリに直接関係しないエミュレータの環境を設定します。
この章で説明する内容は、メニューから「設定」→「エミュレータ環境設定...」を選択したときの各種機能です。

8.1. ビルドの設定

8.1.1. 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。

「iα ppliTool」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 ファイル）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 8-1）。

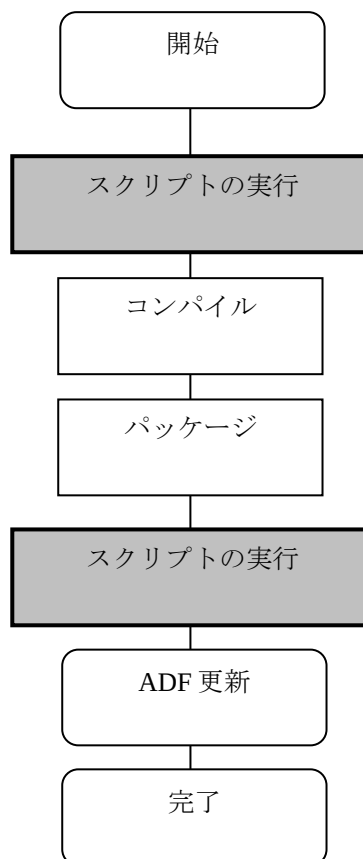


図 8-1 ビルド時のスクリプトコマンド実行の流れ

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 8-2）。

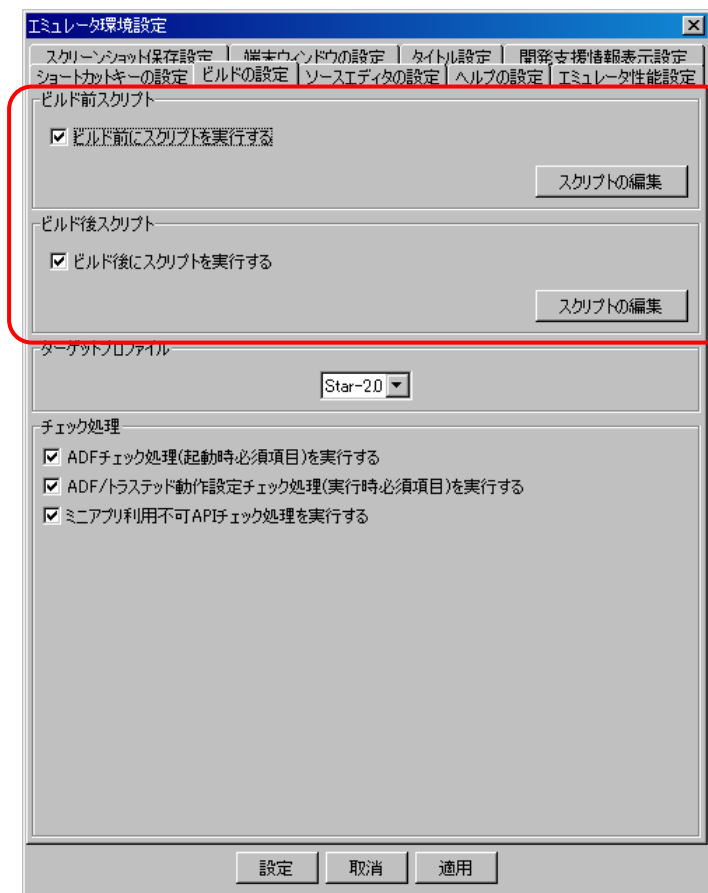


図 8-2 ビルドの設定 - スクリプトの実行

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:¥iDKStar2.0¥lib¥optimize_script1.bat`
- ビルド後スクリプト：
`C:¥iDKStar2.0¥lib¥optimize_script2.bat`

8.1.2. 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 Star のプロファイルを設定します。
デフォルトは、最新の「Star-2.0」になっています。

ビルド時の Star 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 8-3）。

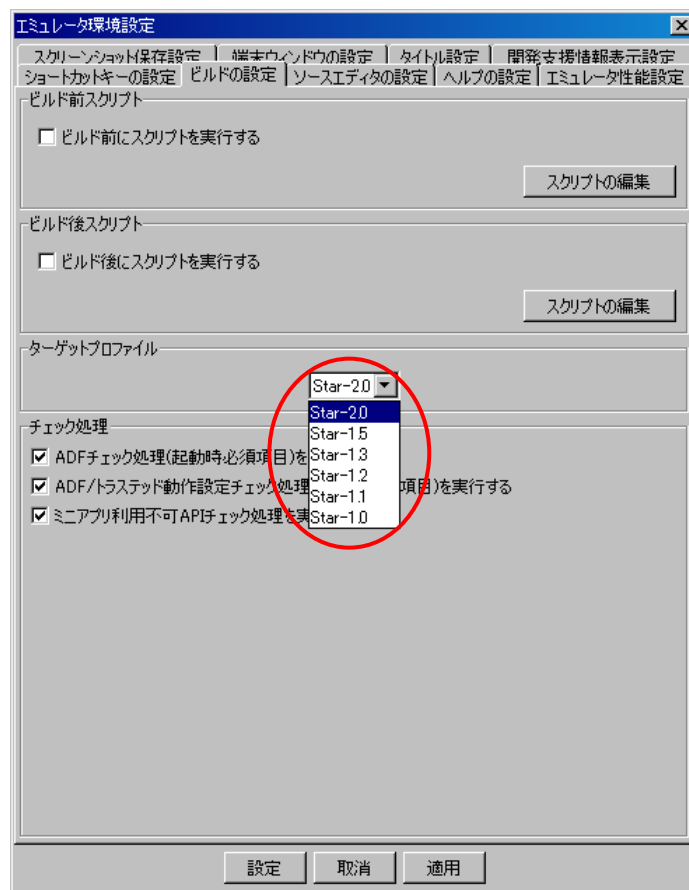


図 8-3 ビルドの設定 - プロファイルの変更

2. ターゲットにしたい Star 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。

ターゲットプロファイル設定にかかわらず、「i α ppli Development Kit for Star-2.0」でビルドした i アプリ（JAR ファイル）は、Star-1.3 プロファイル搭載以降の携帯電話でのみ実行可能です。それ以前の携帯電話では実行できませんのでご注意ください。

8.2. 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、ミニアプリにおいて利用できない API を使っ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 8-4）。

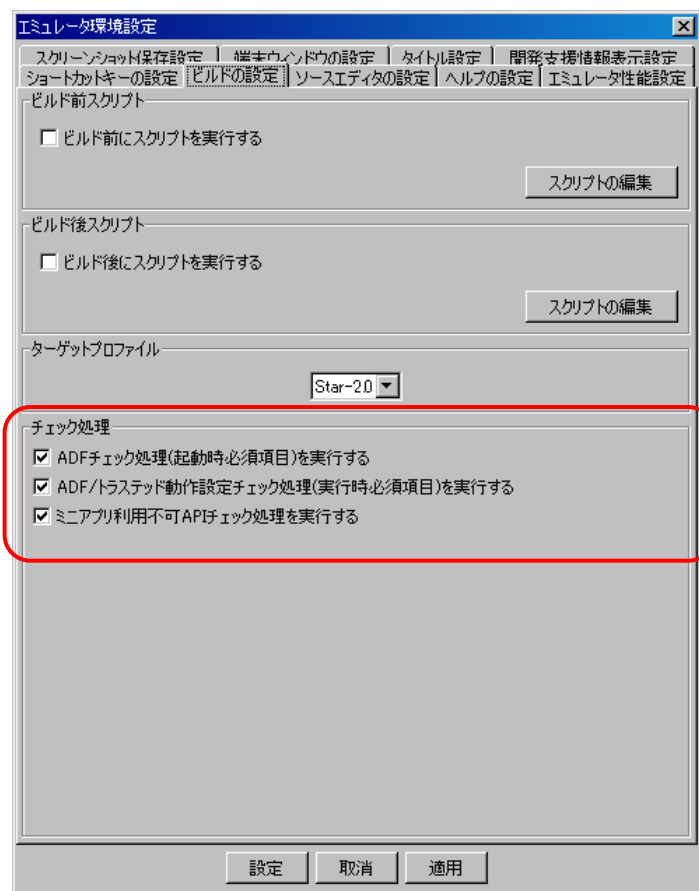


図 8-4 ADF・トラステッド動作設定チェック処理

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 API 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。
ミニアプリのビルド時に ミニアプリで利用できない API を使っていた場合 にビルドを中断したい時は、「ミニアプリ利用不可 API チェック処理を実行する」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 8-5）。

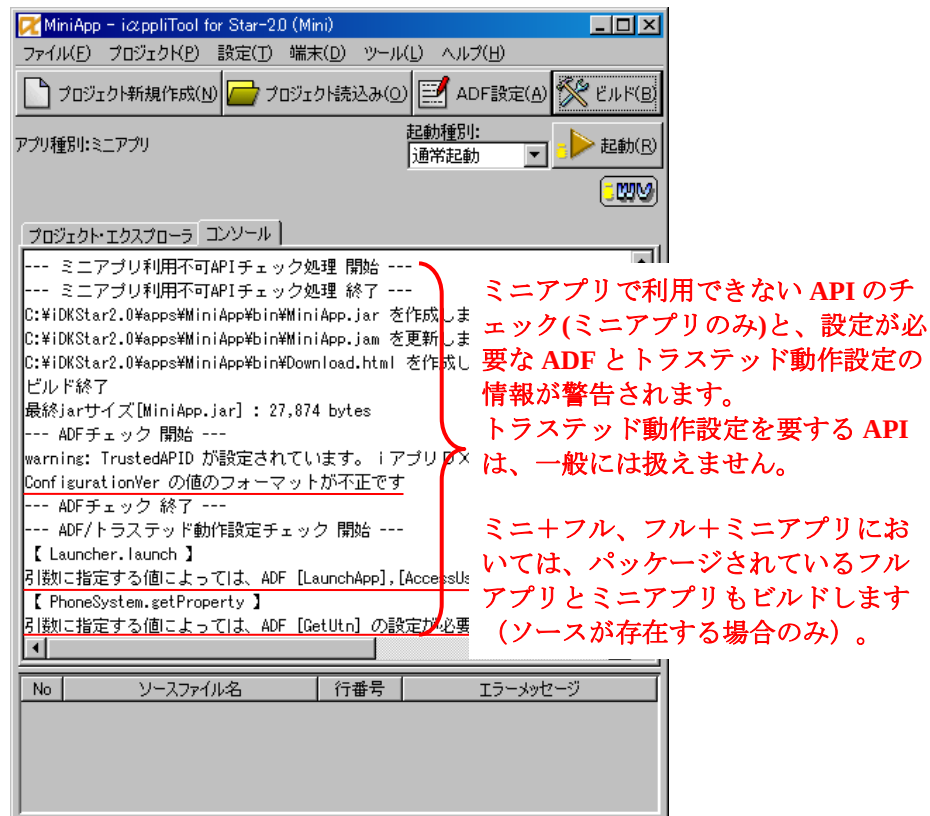


図 8-5 ADF/トラステッド動作設定のチェック結果

8.3. ショートカットキーの設定

「iα ppli Development Kit for Star-2.0」で扱うショートカットキーを設定します。

「iα ppli Development Kit for Star-2.0」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：

■ 「iα ppliTool」

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

■ エミュレーション端末

《12 キーボード設定の端末ウィンドウ選択時》

数字ボタン：	0 ～ 9
# ボタン：	#
* ボタン：	*
決定ボタン：	Enter (↵)
クリアボタン：	Esc
方向ボタン：	Up (↑) , Down (↓) , Left (←) , Right (→)
ファンクションキー1：	A (テキスト編集集中は無効)
ファンクションキー2：	S (テキスト編集集中は無効)
ファンクションキー3：	Z (テキスト編集集中は無効)
ファンクションキー4：	X (テキスト編集集中は無効)

《フルキーボード設定の端末ウィンドウ選択時》

英数字・記号キー：	対応する文字キー
Enter キー：	Enter (↵)
Esc キー：	Esc
方向ボタン：	Up (↑) , Down (↓) , Left (←) , Right (→)

※ 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。

※ フルキーボード設定の端末ウィンドウが選択されている場合は、フルキーボードモードが無効（12 キーボードモードとして動作する場合）であっても、ショートカットキーは《フルキーボード設定の端末ウィンドウ選択時》の設定となります。

「iα ppliTool」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 8-6）。

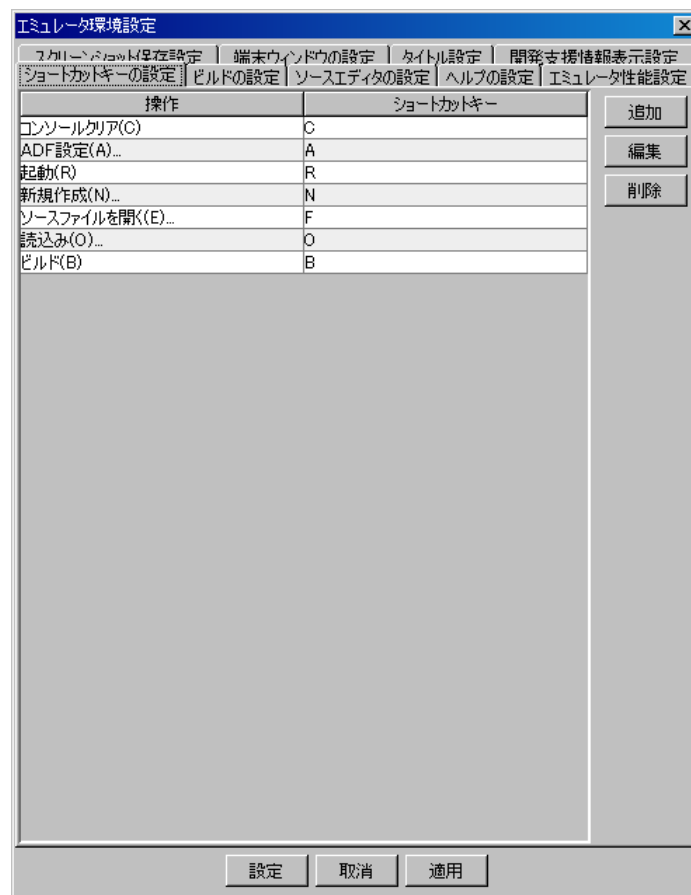


図 8-6 ショートカットキーの設定

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 8-7）。
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 8-7 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 8-1 に示すとおりです。

表 8-1 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: ADF 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 8-8)

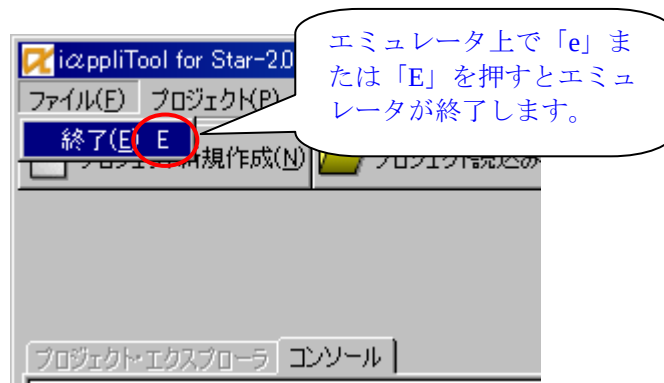


図 8-8 終了メニューに「E」のショートカットキーが設定された状態

8.4. ヘルプの設定

「iα ppliTool」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 8-9）。

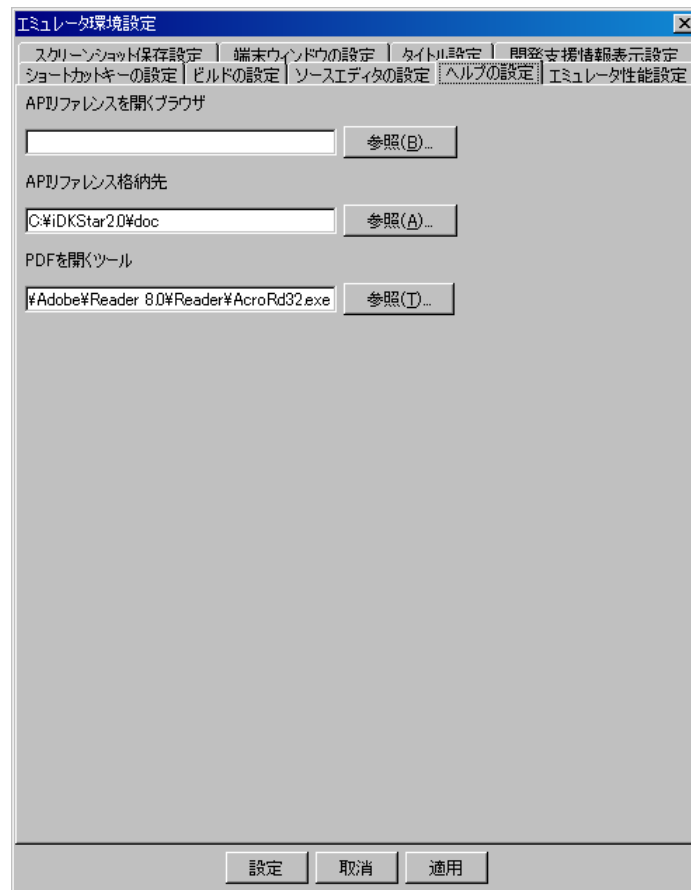


図 8-9 ヘルプの設定画面

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 8-2 に示すとおりです。

表 8-2 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	操作ガイド、リリースノートを参照するためのツールを設定します。 ■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 Adobe Reader

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

8.5. 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電話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 8-10）。

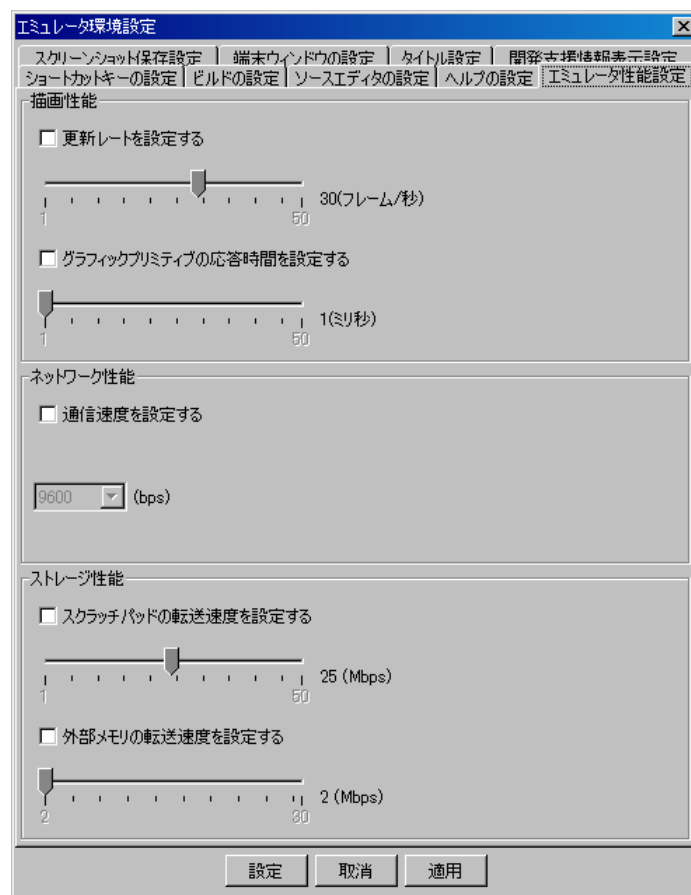
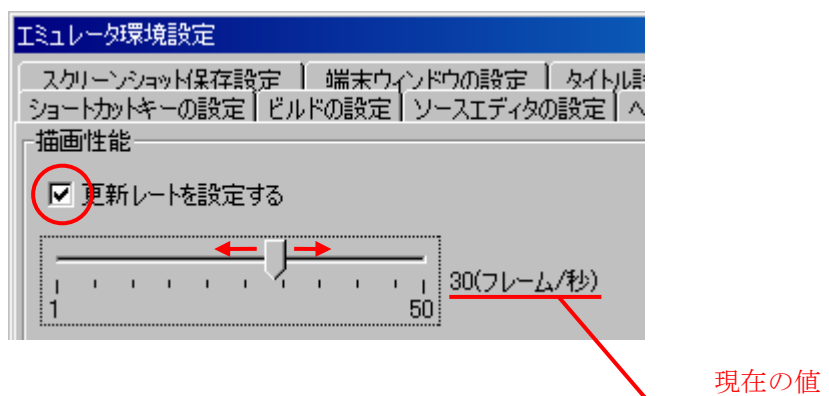


図 8-10 エミュレータ性能設定画面

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ダ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。



各設定項目は 表 8-3 に示すとおりです。

表 8-3 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 1～50fps 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 1～50ms 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 ミ秒。	1 ミ秒
ネットワーク性能	通信速度を設定する	HTTP 通信および、TCP/UDP 機能の待ち時間を bps で設定します。 未設定の場合、PC 性能に依存。	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 1～50Mbps 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 2～30Mbps 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

8.6. エミュレーション端末ウィンドウの設定

フルアプリにおいて、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。^[9]

また、フルアプリ・ミニアプリ共、エミュレータで実装している各オプション機能の無効化の設定が行えます。

端末ウィンドウ設定を追加・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...」→「端末ウィンドウの設定」を選択します（図 8-12）。

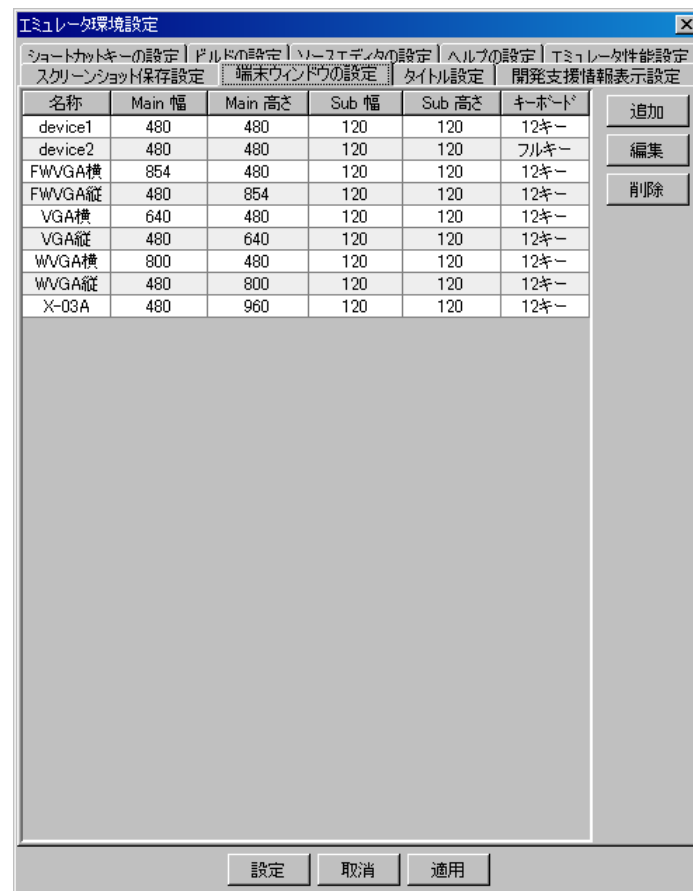


図 8-12 エミュレーション端末ウィンドウ一覧画面

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。
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 8-13）。
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。初期状態では、表 8-4 のエミュレーション端末が設定されています。このうち **device1**、**device2** については編集・削除できませんのでご注意ください。

^[9] Display#getWidth()及び Display#getHeight() は、ここで設定した値と同じになります。
ミニアプリ、Widget View 表示は、必ず WVGA サイズ 480×800 ドットになります。

図 8-13 エミュレーション端末ウィンドウ編集画面

表 8-4 初期状態の端末ウィンドウ設定

名称	メインディスプレイサイズ	備考
device1	幅 480×高さ 480	標準サイズ 設定変更・削除不可
device2	幅 480×高さ 480	標準サイズ（フルキーボード対応） 設定変更・削除不可
FWVGA 横	幅 854×高さ 480	設定変更、削除、オプション機能対応設定の変更が可能です。 （ただし「iα ppli Development Kit for Star-2.0」を再インストールした場合は設定がすべて初期化されます）
FWVGA 縦	幅 480×高さ 854	
VGA 横	幅 640×高さ 480	
VGA 縦	幅 480×高さ 640	
WVGA 横	幅 800×高さ 480	
WVGA 縦	幅 480×高さ 800	

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 8-5 に示すとおりです。

表 8-5 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。 また Windows のファイル名として使用できない文字 (“/”、“*”、“?”など) は使用できません。	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 96 ～ 1024 の範囲で指定します。(※)	480 (最小サポート)
メインディスプレイ 高さ	メインディスプレイの高さを 72 ～ 1024 の範囲で指定します。(※)	480 (最小サポート)
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 96 ～ 240 の範囲で指定します。(※)	120
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 72 ～ 320 の範囲で指定します。(※)	120
キーボード	フルキーボード対応端末か否かを設定します。「フルキーボード」を選択すると、フルキーボード関連の API が使用可能となります。	12 キーボード
立体視 解像度比	平面表示時とステレオ表示時のディスプレイ解像度比を「1」、「2」から選択します。 i アプリオプション API のステレオ表示機能で取得される、解像度比の値となります。 また、画面上のステレオ表示にも影響があります。	2
立体視 深さ 0 の視差オフセット の値	深さ 0 の位置での視差オフセットの値を 0 ～ 99999 の範囲で設定します。 i アプリオプション API のステレオ表示機能で取得される、視差オフセットの値の算出時に利用されます。	0
立体視 深さが 1 変化する時に視 差オフセットが変化する 値	深さが 1 変化する時に視差オフセットが変化する値を 0 ～ 99999 の範囲で設定します。 i アプリオプション API のステレオ表示機能で取得される、視差オフセットの値の算出時に利用されます。	1
立体視 立体視強度	立体視強度を「標準(強)」、「中」、「弱」、「無し」から選択します。 i アプリオプション API のステレオ表示機能で取得される、立体視強度の値となります。	標準(強)

※設定可能な範囲であり、動作を保証する範囲ではありません。正常に動作する範囲はエミュレータを利用する環境（ビデオカード、モニタ等）に依存します。

画面の解像度を幅：W、高さ：Hとした場合、利用可能な範囲は以下ようになります。

メインディスプレイ 幅：96 ～ W - 320 高さ：72 ～ H - 210
サブディスプレイ 幅：96 ～ 240 高さ：72 ～ H - 560

また、ディスプレイサイズを大きくするとヒープメモリが不足する可能性があります。
もし、i アプリの起動に失敗する場合は、「6.2. 全般設定」にてヒープメモリを変更してください。(i モード携帯電話の仕様と同じ値に設定することが望ましいです。)

8.6.1. オプション機能対応設定

携帯電話の機種によって対応有無に差がある機能（オプション機能）について、対応／非対応を設定することができます。

オプション機能の対応／非対応を設定するには：

1. 図 8-13 のエミュレーション端末ウィンドウ編集画面で、「オプション機能対応設定」ボタンを押します。

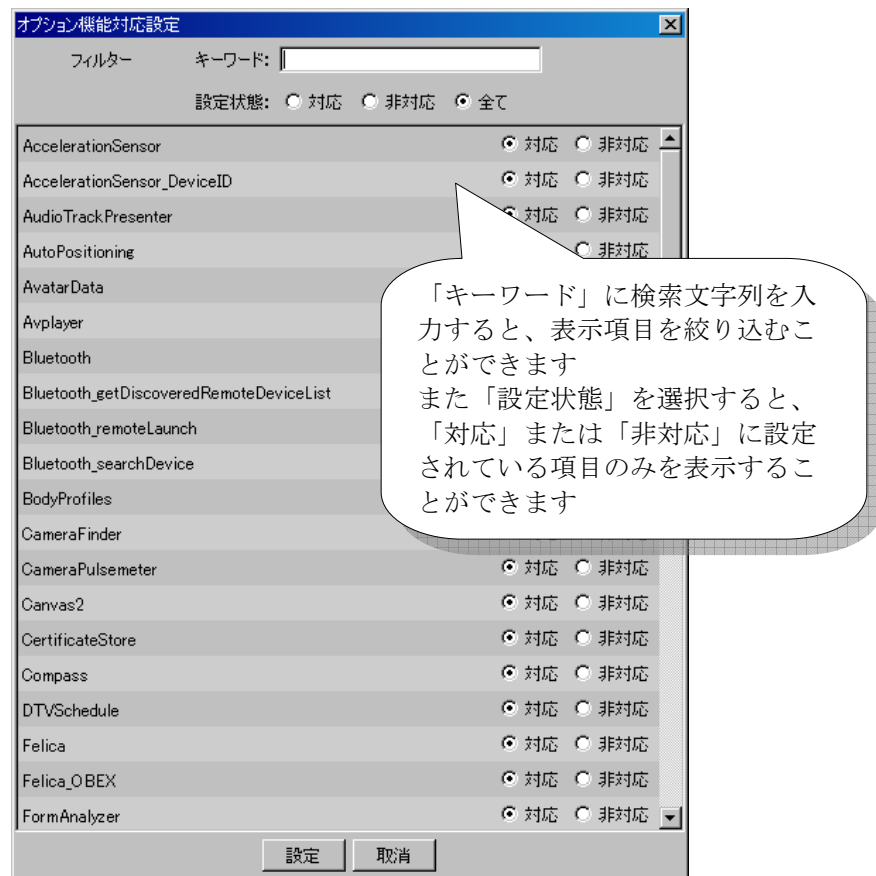


図 8-14 オプション機能対応設定画面

2. オプション機能対応設定画面が表示され、選択した端末ウィンドウ設定についての現在のオプション機能対応設定状態が表示されます。
初期状態では、すべて「対応」に設定されてますので、非対応としたい機能について「非対応」を選択し、「設定」を押してください。

3. エミュレーション端末ウィンドウ編集画面で「設定」を押します。
4. エミュレーション端末ウィンドウ一覧画面で「設定」または「適用」を押します。

オプション機能対応設定画面に表示される機能は表 8-6 の通りです。なお、「依存関係」に他項目名が記載されている項目はそれ単独で「対応」に設定しても無効で、併せて「依存関係」に記載した項目を「対応」に設定する必要があります。

表 8-6 オプション機能設定項目

設定項目	説明	依存関係
AccelerationSensor	加速度センサ機能 com.docomostar.device.location.AccelerationSensor	
AccelerationSensor_DeviceID	加速度センサ機能（デバイス指定による加速度センサ操作） com.docomostar.device.location.AccelerationSensor.getAccelerationSensor(int)	AccelerationSensor
AudioTrackPresenter	動画のオーディオトラックのみの再生 com.docomostar.ui.AudioPresenter.getAudioTrackPresenter	
AutoPositioning	オート GPS com.docomostar.system.AutoPositioning	
AvatarData	アバターデータ機能 com.docomostar.system.AvatarData	
Avplayer	音楽プレイヤー機能 com.docomostar.media.avcontent com.docomostar.ui.avplayer	
Bluetooth	Bluetooth 機能 com.docomostar.device.Bluetooth	
Bluetooth_getDiscoveredRemoteDeviceList	Bluetooth 機能（登録済み機器の一括取得） com.docomostar.device.Bluetooth.getDiscoveredRemoteDeviceList	Bluetooth
Bluetooth_remoteLaunch	Bluetooth 外部機器からのアプリケーション起動要求 com.docomostar.StarEventObject.STAR_BLUETOOTH_CONNECT_REQUEST_RECEIVED	Bluetooth
Bluetooth_searchDevice	Bluetooth 機能（searchDevice/startPairing メソッドによる機器検索および機器登録） com.docomostar.device.Bluetooth	Bluetooth
BodyProfiles	身体情報アクセス機能 com.docomostar.opt.system.BodyProfiles	
CameraFinder	カメラファインダ機能 com.docomostar.opt.device.CameraFinder	
CameraPulsemeter	脈拍計機能 com.docomostar.opt.device.CameraPulsemeter	
Canvas2	Canvas 座標系変更機能 com.docomostar.opt.ui.Canvas2	
CertificateStore	ユーザ証明書へのアクセス com.docomostar.security.PKCS7Signer.sign(int) com.docomostar.system.CertificateStore.selectEntryId	

設定項目	説明	依存関係
Compass	電子コンパス機能 com.docomostar.device.location.Compass	
DTVSchedule	DTV 視聴/録画予約スケジューラ機能 com.docomostar.system.DTVSchedule	Launch_DTV
Felica	FeliCa 機能 com.docomostar.StarEventObject.STAR_FELICA_PUSHED com.docomostar.device.Bluetooth.startPairingByOB com.docomostar.device.felica	
Felica_OBEX	FeliCa アドホック通信による OBEX 通信機能 com.docomostar.StarEventObject.STAR_FELICA_ADHOC_REQUEST_RECEIVED com.docomostar.io.FelicaClientObexConnection com.docomostar.io.FelicaServerObexConnection	Felica
FormAnalyzer	フォーム分析機能 com.docomostar.opt.system.formanalyzer	
FullKeyboard	フルキーボード機能 com.docomostar.opt.ui.FullKeyboard	
FullScreen	アプリの全画面表示 com.docomostar.ui.Frame.setSoftLabelVisible	
GestureReader	動作認識機能 com.docomostar.device.gesturereader	
GPS	GPS 機能 com.docomostar.device.location.LocationProvider com.docomostar.system.LocationRecord	
Graphics_drawScaledRawImage	RAW 画像の拡大・縮小描画 com.docomostar.ui.Graphics.drawScaledRawImage	RawImageCapture
Graphics2_drawNthImage	アニメーション GIF 画像の表示 com.docomostar.opt.ui.Graphics2.drawNthImage	
Graphics2_getSyncUnlock	表示デバイスの垂直同期に合わせた描画 com.docomostar.opt.ui.Graphics2.getSyncUnlockInterval com.docomostar.opt.ui.Graphics2.syncUnlock	
GraphicsOGL_flushAndStoreCommands	OpenGL 描画の非同期実行 com.docomostar.ui.opengl.GraphicsOGL.flushAndStoreCommands	
GyroSensor	ジャイロセンサ・基本機能 com.docomostar.device.location.GyroSensor	
GyroSensor_Option	ジャイロセンサ・オプション機能 com.docomostar.opt.device.GyroSensor	
GyroSensor_Option2	ジャイロセンサ・オプション機能 (GyroSensor2) com.docomostar.opt.device.location.GyroSensor2	GyroSensor
ImageStore_addEntry	ネイティブ画像データ管理 (画像データの一括登録) com.docomostar.system.ImageStore.addEntry(MediaImage[])	
ImageStore_selectEntryId	ネイティブ画像データ管理 (画像データのエントリ ID 取得) com.docomostar.system.ImageStore.selectEntryId	

設定項目	説明	依存関係
IrReceiver	赤外線受信機能 com.docomostar.opt.device.IrReceiver	
IrRemoteControl	ミニアプリの赤外線リモコン機能対応 com.docomostar.device.IrRemoteControl	
IrSimple	IrSimple 通信による OBEX 通信機能 com.docomostar.io.IrClientObexConnection com.docomostar.io.IrServerObexConnection	
Launch_Browser_Suspend	アプリサスペンドによるブラウザ起動 com.docomostar.system.Launcher.LAUNCH_BROWSER_SUSPEND	
Launch_DTV	DTV 連携機能 com.docomostar.StarEventObject.STAR_CALLED_BY_DTV com.docomostar.system.Launcher.LAUNCH_DTV	
LimitInputSize	IME、テキストボックスの入力文字数制限 com.docomostar.ui.Canvas.imeOn(String, int, int, int) com.docomostar.ui.TextBox.setInputSize	
LocationRecord	位置履歴データ取得 com.docomostar.system.CityInfoStore com.docomostar.system.LocationRecord	GPS
MailAgent_getRemainingBytes	メール機能（メール本文の残バイト数の取得） com.docomostar.system.MailAgent.getRemainingBytes	
MailAgent2	拡張ネイティブメーラ連携機能 com.docomostar.opt.system.AttachmentData com.docomostar.opt.system.MailAgent2	
MascotCapsule_eruptionST	Eruption パッケージによる 3D 描画機能 com.docomostar.opt.ui.eruption	
MovieStore_getEntry	動画データ保存領域へのアクセス機能（getEntry メソッドのサポート） com.docomostar.system.MovieStore.getEntry	
MovieStore_selectEntry	動画データ保存領域へのアクセス機能（selectEntry メソッドのサポート） com.docomostar.system.MovieStore.selectEntry	
Pedometer	歩数計・基本機能 com.docomostar.system.Pedometer	
Pedometer_Option	歩数計・オプション機能 com.docomostar.opt.device.Pedometer	
PhoneTheme_setMenuIcons	テーマ設定（メニューアイコン画像の設定） com.docomostar.system.PhoneTheme.setMenuIcons	
PhysicalActivityMonitor	身体活動量計・基本機能 com.docomostar.system.PhysicalActivityMonitor	
PhysicalActivityMonitor_Option	身体活動量計・オプション機能 com.docomostar.opt.device.PhysicalActivityMonitor	
PointingDevice_Joystick	ポインティングデバイス機能（ジョイスティックモード） com.docomostar.opt.ui.PointingDevice	

設定項目	説明	依存関係
PointingDevice_Mouse	ポインティングデバイス機能（マウスモード） com.docomostar.opt.ui.PointingDevice	
PointingDevice_OnFlashPlayer	Flash コンテンツ再生中のポインティングデバイス	PointingDevice_Mouse
RawImageCapture	RAW 画像取得機能 com.docomostar.device.RawImageCapture	
RawImageCapture_getRawImageAngleSize	RAW 画像取得機能（画角の取得） com.docomostar.device.RawImageCapture.getRawImageAngleSize	RawImageCapture
Security_Cipher	暗号化機能 com.docomostar.security.RandomNumberGenerator com.docomostar.security.CipherPBE	
Sound3d	3D サウンド機能 com.docomostar.ui.sound3d	
SpeechSynthesizer	音声合成機能 com.docomostar.opt.ui.SpeechSynthesizer	
StereoScope	ステレオ表示機能 com.docomostar.opt.ui.stereoscope	
Synthesizer	シンセサイザ・基本機能 com.docomostar.ui.Synthesizer	
Synthesizer_Option	シンセサイザ・オプション機能 com.docomostar.opt.ui.Synthesizer	
ThermoHygrometer	ネイティブ温湿度計機能 com.docomostar.opt.system.ThermoHygrometer	
ThermoHygrosensor	温湿度センサ機能 com.docomostar.opt.device.ThermoHygroSensor	
TouchDevice	タッチパネルデバイス・基本機能 com.docomostar.device.TouchDevice	
TouchDevice_OnFlashPlayer	Flash コンテンツ再生中のタッチパネルデバイス	TouchDevice または TouchDevice_Option
TouchDevice_Option	タッチパネルデバイス・オプション機能 com.docomostar.opt.ui.TouchDevice	
UART	UART 機能 Connector.open("comm:")	
WalkingNavigation	歩行方位取得機能 com.docomostar.device.location.WalkingNavigation com.docomostar.StarEventObject.STAR_WALKING_NAVIGATION_STOPPED	

8.7. タイトル設定

「iα ppliTool」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、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。

「iα ppliTool」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 8-15）。



図 8-15 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。

8.8. 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、「9.6. 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 8-16）。

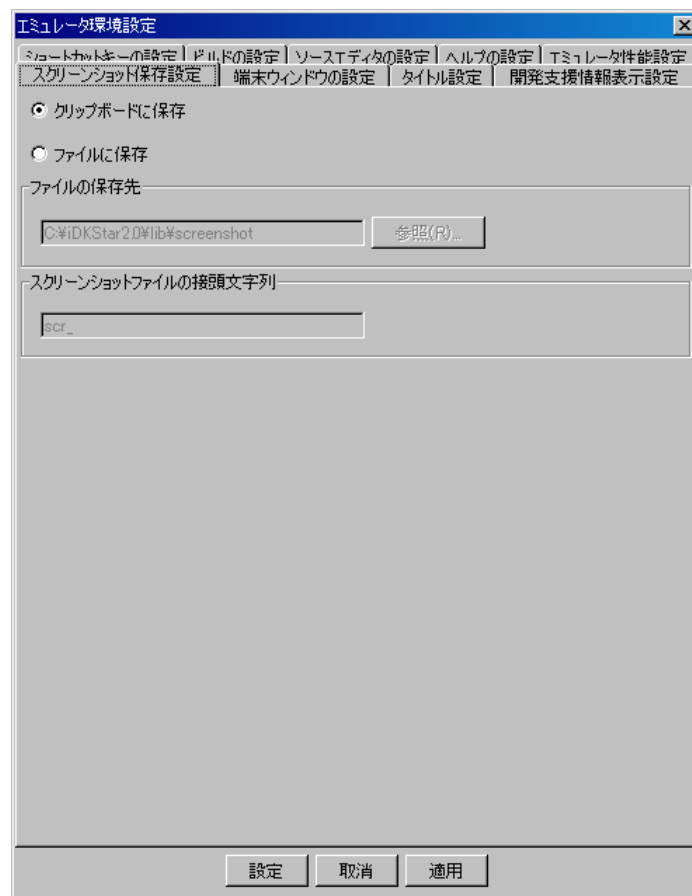


図 8-16 スクリーンショット保存設定

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、「クリップボードに保存」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文字列を指定します。
ファイルは、「[接頭文字列]XXXXXXXXX.bmp」(XXXXXXXXX:連番) で保存されます。
4. 「設定」もしくは「適用」を押します。

8.9. 開発支援情報表示設定

開発支援情報画面の表示方法を変更することができます。

開発支援情報画面の詳しい説明については、「8.16.3 開発支援情報」を参照してください。

開発支援情報画面の表示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...」→「開発支援情報表示設定」を選択します（図 8-17）。

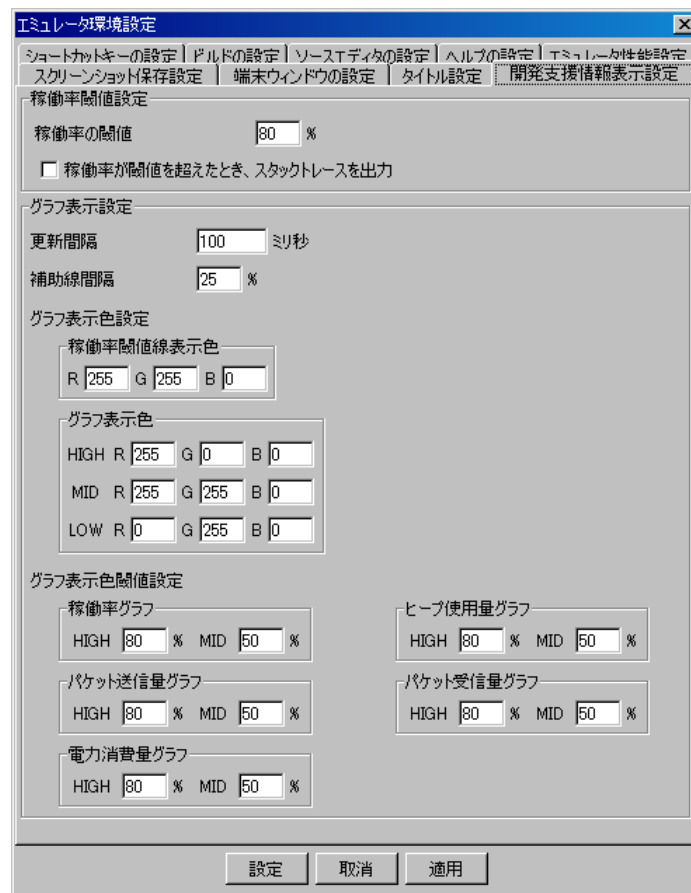


図 8-17 開発支援情報表示設定

2. 各設定項目を設定して、「設定」もしくは「適用」を押します。

各設定項目は表 8-7 に示すとおりです。

表 8-7 開発支援情報表示の設定項目

設定項目	説明	デフォルト
稼働率の閾値	稼働率の閾値を 0～100%の範囲で設定します。	80

設定項目	説明	デフォルト
稼働率が閾値を超えたとき、スタックトレースを出力	稼働率が閾値を超えたときに、コンソールへスタックトレースを出力する場合はチェックボックスを ON にします。	OFF
更新間隔	各グラフの更新間隔を 10～99999 ミリ秒の範囲で設定します。	100
補助線間隔	各グラフの補助線の間隔を 5～100% の範囲で設定します。	25
稼働率閾値線表示色	稼働率グラフに表示する閾値線の表示色 (R、G、B) をそれぞれ 0～255 の範囲で設定します。	R:255、G:255、B:0
グラフ表示色	各グラフの表示値が高域 (HIGH) / 中域 (MID) / 低域 (LOW) の時の、グラフの表示色 (R、G、B) をそれぞれ 0～255 の範囲で設定します。	(HIGH) R:255、G:0、B:0 (MID) R:255、G:255、B:0 (LOW) R:0、G:255、B:0
グラフ表示色閾値設定	グラフごとに、高域 (HIGH) と判定する値、中域 (MID) と判定する値を設定します。それぞれ 0～100% の範囲で設定します。ただし HIGH は MID よりも大きい値を設定してください。	(稼働率グラフ) HIGH:80、MID:50 (ヒープ使用量グラフ) HIGH:80、MID:50 (パケット送信量グラフ) HIGH:80、MID:50 (パケット受信量グラフ) HIGH:80、MID:50 (電力消費量グラフ) HIGH:80、MID:50

8.10. FeliCaPush 通知パラメータ設定

FeliCaPush 時の通知パラメータを設定することができます。

FeliCaPush の通知パラメータを設定するには：

1. メニューから「設定」→「FeliCaPush 通知パラメータ設定...」を選択します。
2. FeliCaPush通知パラメータ設定画面が表示されるので、通知パラメータを設定して「設定」を押します（図 8-18）。

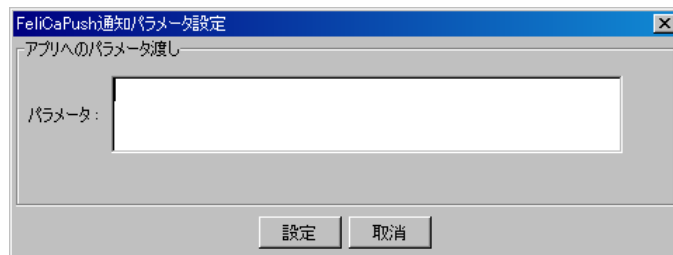


図 8-18 FeliCaPush 通知パラメータ設定画面

8.11. 起動時にプロジェクトを読み込む

「iα ppliTool」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「iα ppliTool」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。

8.12. jam に関連付ける

ADF (.jam ファイル) を「iα ppliTool」に関連付けします。これにより、.jam ファイルをクリックするだけで、i アプリの場合は、「iα ppliTool」が起動し、i アプリが実行されます。なお、i アプリ Bookmark の場合は、「iα ppliTool」が起動します。(実行はされません。)

ADF (.jam ファイル) を「iα ppliTool」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。

※上記の操作によりファイルの関連付けの設定・解除を行うためには、「iα ppliTool」が管理者権限を持つユーザによって起動されている必要があります。さらに、Windows 7 環境においては「管理者として起動」を選択して起動されている必要があります。起動方法は「第 3 章 iα ppliTool の起動」を参照ください。

8.13. 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^[10]。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。

^[10] 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。

8.14. 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。

本機能は、フルアプリにおいてのみ有効です。ミニアプリまたは **Widget View** 表示では、端末の選択に関係なく、必ず **WVGA サイズ (480×800 ドット)** になります。またフルキーボード関連 **API** は使用できません。

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 8-19）。

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「i α 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。

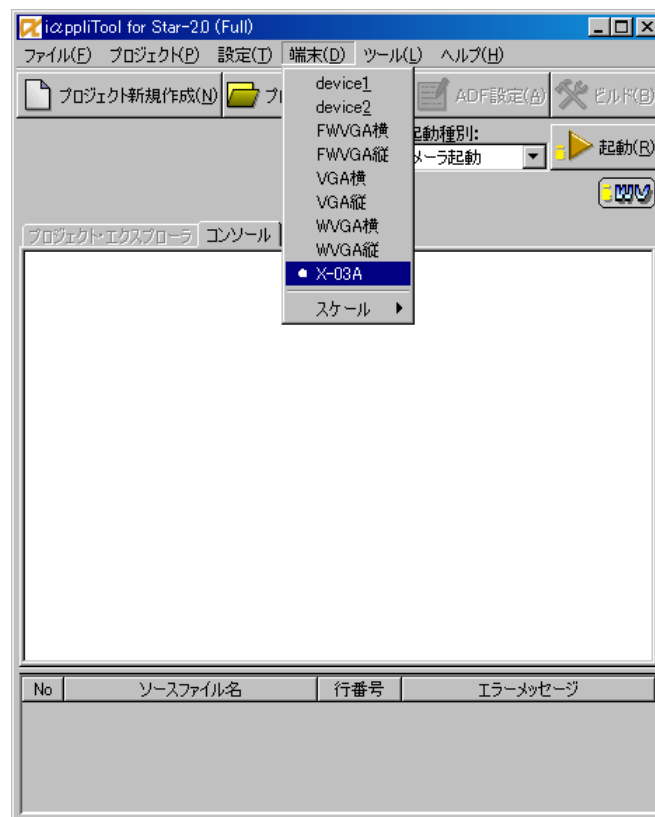


図 8-19 端末の選択

 ワンポイント

エミュレーション端末は、携帯電話の機種に合わせて異なるウィンドウサイズ、異なるキーボード設定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、「8.6. 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。

8.15. エミュレーション端末のスケール変更

WVGAサイズ (480×800 ドット)対応に伴い、「iα ppliTool」では、エミュレーション端末の表示スケールを変更することができます。^[11]

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」→「スケール」を選択し、スケールの割合を選択します（図 8-20）。
このとき、スケール 100% を選択した場合は表示縮小処理を行いません。

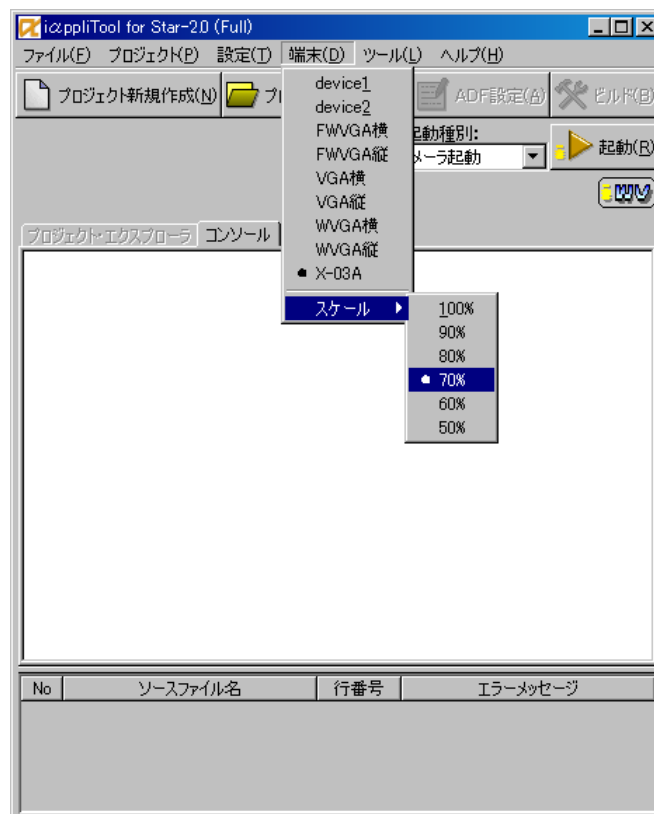


図 8-20 端末ウィンドウのスケール変更

^[11] スケールを変更した場合、アプリ実行中に表示縮小処理が実行されるため、多少、処理が遅くなります。表示処理の速度は、お使いの PC のマシンスペックにより異なります。

8.16. 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 8-21）。

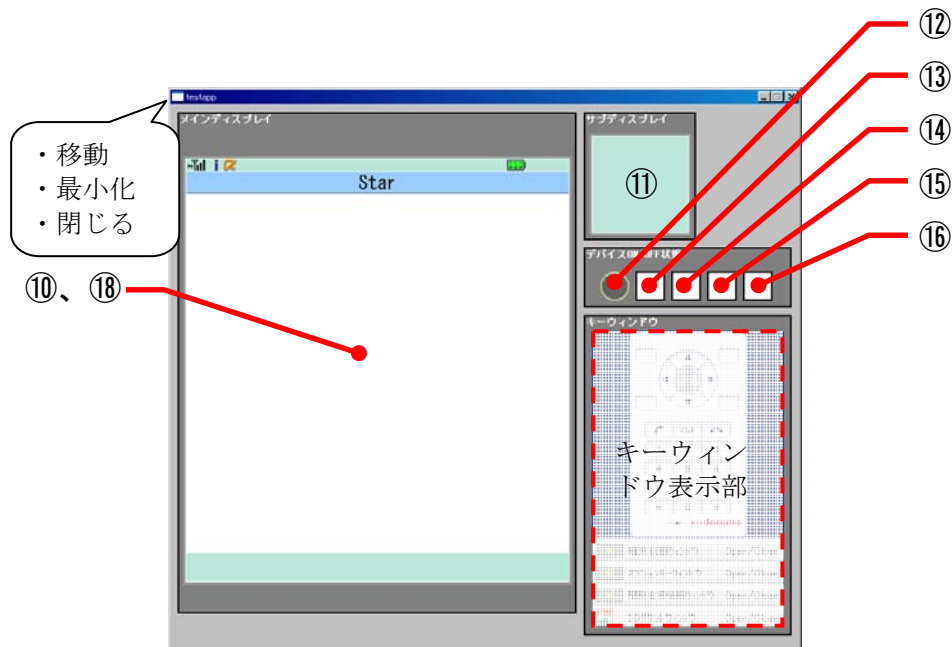
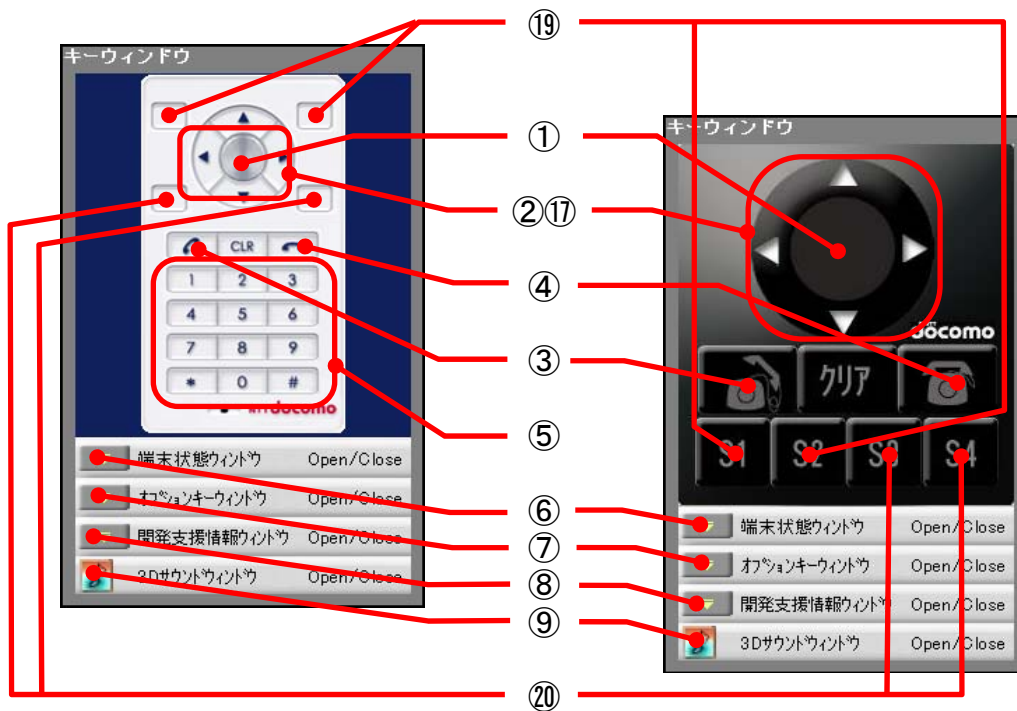


図 8-21 エミュレーション端末

エミュレーション端末右下部の「キーウインドウ」表示部分は、「8.14 エミュレーション端末の選択」で選択されている端末のキーボード設定によってボタンレイアウトが切り替わります。（図 8-22）

キーボード設定については「8.6 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。



《12 キーボード対応端末》

《フルキーボード対応端末》

図 8-22 エミュレーション端末（キーウィンドウ表示部）

- ①. 決定ボタン
12 キーボード端末の場合、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。
フルキーボードの場合、このボタンに対応するキーはキーボードにはありません。
- ②. 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。
- ③. 発信ボタン
通常の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。
- ④. 終話ボタン
③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、通常のアプリケーションおよび Widget View を強制終了させます。
- ⑤. 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。

このボタンは、フルキーボード対応端末にはありません。フルキーボード対応端末が選択されている場合は、該当する文字（英字、数字、記号等）に対応するキーボードのキーを使用します。

⑥. 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、「8.16.1. 端末状態ウィンドウの操作」で詳しく説明します。

⑦. 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、「8.16.2. オプションキーウィンドウの操作」で詳しく説明します。

⑧. 開発支援情報画面表示

i アプリ実行中に稼働率、ヒープ使用量等をグラフ表示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。

開発支援情報画面については、「8.16.3 開発支援情報」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、「8.16.4. 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。

さらに、ステレオ表示が有効な場合は、ステレオ表示方式の切替もできます。

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、ステレオ表示方式には、それぞれ以下に示す種類があります：

■ 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」

- 内部電源
- 外部電源
- 「バッテリー残量」
 - 1～100
- 「電波状態」
 - アンテナ 1 本
 - アンテナ 2 本
 - アンテナ 3 本
- 「エリア状態」
 - 圏内 (HSDPA+HSUPA)
 - 圏内 (HSDPA)
 - 圏内 (FOMA)
 - 圏内 (ローミングアウト)
 - 圏外
 - セルフモード
- 「マナーモード状態」
 - マナーモード OFF
 - マナーモード ON
- 「UIM ロック状態」 (i アプリ起動中は設定不可)
 - 通常 (VERSION 1)
 - 通常 (VERSION 2)
 - 制限中
- 「歩数計状態」
 - 停止中
 - 計測中
- ※アプリから歩数計の状態を変更すると、その結果が反映されます
- 「身体活動量計状態」
 - OFF
 - ON
- ※アプリから身体活動量計の状態を変更すると、その結果が反映されます
- 「脈拍計測定状態」
 - 正常
 - 暗い
 - 明るい
 - 位置不正
 - 未検出
- 「加速度センサ状態」
 - デバイス 1
 - OFF
 - ON
- ※加速度センサの「デバイス 0」は常に ON であり変更できません
- 「端末音声出力先状態」
 - 内部出力
 - 外部出力
- 端末ウィンドウ以外で発生できるイベント
 - 「タッチパネル」 (タッチデバイスオプション API 用)
 - 「広げる」
 - 「つまむ」
 - 「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」
- 「ダブルタップイベント」
 - 「背面」
 - 「LCD 面」
 - 「左側面」
 - 「右側面」
- 「招集イベント」
- 「電子コンパスイベント」
 - 「キャリブレーション要求」
 - 「キャリブレーション完了」
- 「ジャイロセンサイベント」
 - 「キャリブレーション要求」
 - 「キャリブレーション中断」
 - 「キャリブレーション完了」
- 「音楽コンテンツ要更新イベント」
- 「FeliCa 外部接続待ち」

このメニューを選択すると、コンソール上に「FeliCa 外部接続待機開始しました」と表示され、現在起動している i アプリで外部接続を待ち受けます。

「FeliCa 外部接続待機終了しました」と表示されるまでの間、相手からの接続を待ち受けます。
- 「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。StarApplication#addEventListener にて StarEventObject.STAR_FELICA_PUSHED を指定して StarEventListener が登録されている場合は、リスナに FelicaPushedEvent 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ことができます。
- 「外部 HDP 機器からの接続」
 - 「接続」

このメニューを選択すると、外部 HDP 機器選択画面が表示されます。いずれか一つを選択し「確定」を押すことで、外部 HDP 機器からの接続要求イベントをエミュレーションします。

 - Bluetooth#scan を実行中の場合、選択画面には Bluetooth HDP 設定画面から設定された全ての外部 HDP 機器が表示されます。

「確定」を押すと、Inquiry Scan / Page Scan を終了し、機器登録後、選択された外部 HDP 機器を示す RemoteDevice オブジェクトを返却します。
 - StarApplication#addEventListener にて StarEventObject.STAR_BLUETOOTH_CONNECT_REQUEST_RECEIVED を指定して StarEventListener が登録されている場合、選択画面には機器登録済みの Bluetooth アドレスに該当する外部 HDP 機器だけが表示されます。

「確定」を押すと、リスナに BluetoothConnectionEvent が通知

され、選択された外部 HDP 機器を示す DeviceDiscoveryResult オブジェクトを取得できます。

- 「Bluetooth 連携起動」の外部 Bluetooth 機器接続待ち状態の場合、選択画面には事前に StarApplicationManager#addRemoteLauncher によって起動可能外部機器として登録済みの Bluetooth アドレスに該当する外部 HDP 機器だけが表示されます。
「確定」を押すと、Page Scan を終了し、i アプリ起動後に BluetoothConnectionEvent オブジェクトが取得できます。これを用いて選択された外部 HDP 機器を示す DeviceDiscoveryResult オブジェクトを取得できます。

➤ 「中断」

このメニューを選択すると、ネイティブ Bluetooth アプリケーションの Page Scan 中断イベントをエミュレーションします。

StarApplication#addEventListener にて StarEventObject.

STAR_BLUETOOTH_CONNECT_REQUEST_RECEIVED を指定して

StarEventListener が登録されている場合は、リスナに

BluetoothConnectionEvent が通知され、getStatus は STATUS_INTERRUPTED を返却します。

- 「メインスクリーンの変化」

➤ 0(度)

➤ 90(度)

➤ 180(度)

➤ 270(度)

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。

- 「ディスプレイ状態」

➤ 通常

➤ 右 90 度回転

➤ 左 90 度回転

➤ リバース

このメニューを選択すると、ディスプレイの状態の変化を

Canvas#processEvent(int type, int param)に通知します。また、選択されたメニューに応じてメインスクリーンの描画領域が回転します（ただし「リバース」は「通常」と同様の表示となります）。

■ ステレオ表示方式変更

- 「サイドバイサイド表示」

➤ 平行法

➤ 交差法

ステレオ表示が有効な場合にこのメニューを選択すると、選択に応じて平行法もしくは交差法でのステレオ表示に切り替わります。

- 「立体視拡大率切替」

➤ 等倍表示

➤ 横 2 倍拡大表示

ステレオ表示が有効な場合、かつ解像度比が 2 に設定されている場合にこのメニューを選択すると、選択に応じてディスプレイのサイズが等倍表示もしくは横 2 倍拡大表示に切り替わります。

- 「3D 表示」

ステレオ表示が有効な場合にこのメニューを選択すると、3D 表示（全画面表示）に切り替わります。

i アプリから端末状態を取得するには、`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、これらの状態を取得する場合、ADFの設定項目「`GetSysInfo`」を設定しておく必要があります（「4.8. 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

Star では、サブディスプレイには対応していません。

⑫. ピクチャライト

Star では、ピクチャライトには対応していません。

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. UIM ロック

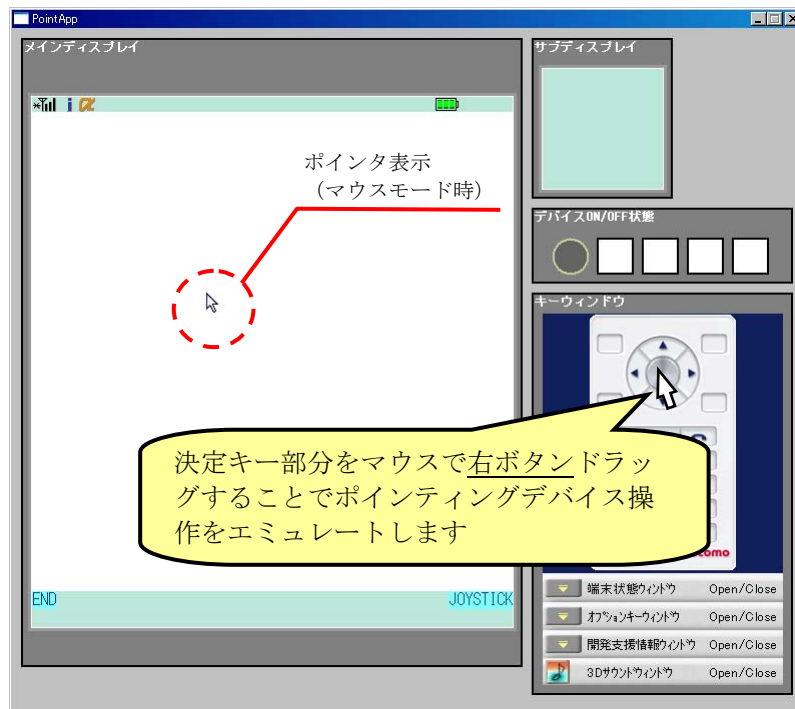
UIM ロック状態を制限中に表示されます。



⑰. ポインティングデバイスの実行

i アプリオプション API 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
キーウィンドウの決定キー部分にマウスカーソルを合わせ、右ボタンを押しながらドラッグ操作することで移動量が取得できます。
- マウスモード
キーウィンドウの決定キー部分にマウスカーソルを合わせ、右ボタンを押しながらドラッグ操作することで LCD にポインタが現われて移動量が取得できます。



⑱. タッチデバイスの実行

タッチデバイス API (TouchEvent) を使った場合に有効になります。

タッチデバイス API には Star-2.0 仕様の基本 API

(com.docomostar.device.TouchDevice) と旧 Star 仕様互換のオプション API

(com.docomostar.opt.ui.TouchDevice) の 2 種類があり、使用方法や発生するイベントが異なります。

- タッチ操作

アプリ描画領域 (DrawArea) 内でマウスの左ボタンを押すとタッチ操作イベント (基本 API の場合は TouchEventObject.TOUCH_PRESSED イベント、オプション API の場合は Display.TOUCH_PRESSED_EVENT イベント) が通知され、画面上の位置座標を取得できます。

- ムーブ操作

タッチ操作後、アプリ描画領域内でマウスの左ボタンを押したままマウスポインタを移動 (ドラッグ) します。

基本 API の場合は、移動している間、一定周期で

TouchEventObject.TOUCH_MOVED イベントが通知され、その時の画面上の位置座標を取得できます。

オプション API の場合は、移動開始時に

Display.TOUCH_MOVEDSTART_EVENT が、移動停止時に

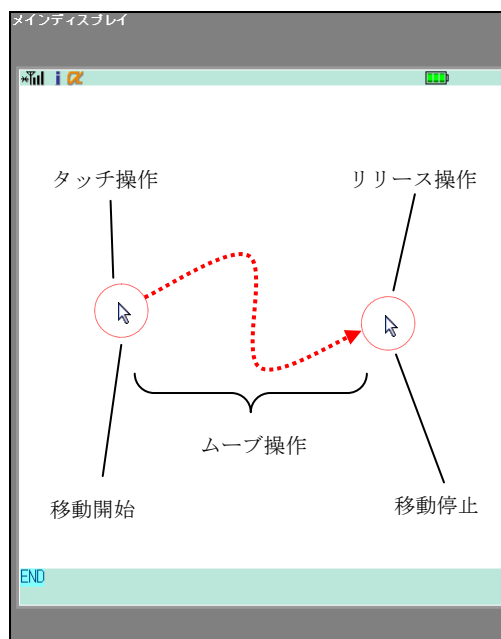
Display.TOUCH_MOVEDEND_EVENT がそれぞれ通知されます。移動中は画面上の最新の位置情報を取得することができます。

- リリース操作

アプリ描画領域内においてタッチ操作やムーブ操作状態からマウスの左ボタンを離した場合にリリース操作イベント (基本 API の場合は

TouchEventObject.TOUCH_RELEASED イベント、オプション API の場合は

Display.TOUCH_RELEASED_EVENT イベント) が通知されます。

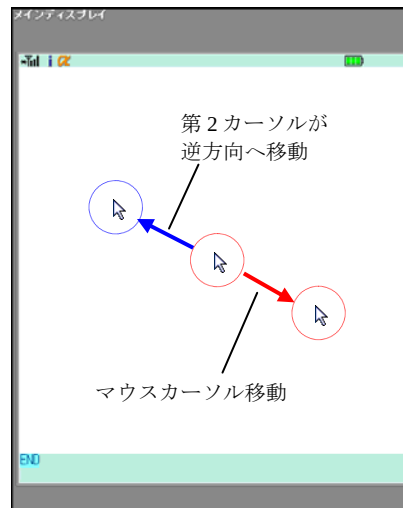


《マルチタッチ操作》※タッチデバイス基本 API のみ対応

本エミュレータでは、2点（2本指）までのマルチタッチ操作をエミュレートできます。

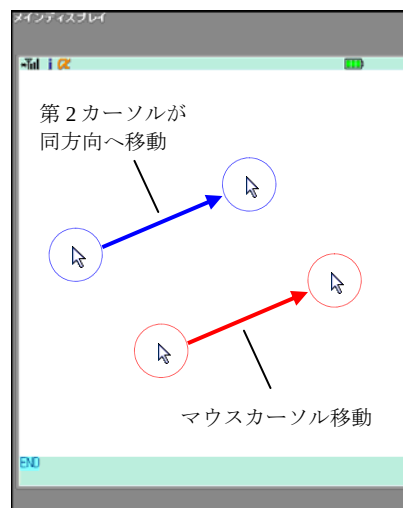
■ 第2カーソルの出現・移動（1）

タッチデバイスが有効な状態で、LCD画面内でキーボードの「F1」キーを押しながらマウスカーソルを移動させると、マウスカーソルの位置に2番目のカーソル（以下「第2カーソル」と表記します）が出現します。そのまま（「F1」キーを押したまま）マウスカーソルを移動すると、マウスカーソルの移動方向と逆方向に第2カーソルが移動します。



■ 第2カーソルの移動（2）

キーボードの「F1」キー、「F2」キーを両方同時に押しながらマウスカーソルを移動すると、マウスカーソルの移動方向と同方向に第2カーソルが移動（平行移動）します。



■ 第2カーソルの消滅

「F1」キーを離すと、第2カーソルが消滅します。

- 第2カーソル位置からのイベント発生
第2カーソルが表示されていて、かつマウスカーソルがLCD画面内にある状態で、タッチ操作・ムーブ操作・リリース操作を行うと、マウスカーソル位置と第2カーソル位置の両方からそれぞれ TOUCH_PRESSED、TOUCH_MOVED、TOUCH_RELEASED イベントが同時に通知されます。

タッチデバイスオプションAPIが有効の場合、第2カーソルは無視されます。オプションAPIのマルチタッチイベント（「広げる」、「つまむ」）を発生させるためには、LCDの右クリックメニュー（⑩参照）を使用してください。

⑩. ファンクションキー1（左側上段）、ファンクションキー2（右側上段）

12 キーボード端末の場合、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、テキスト編集では無効です）。
フルキーボードの場合、このボタンに対応するキーはキーボードにはありません。

⑪. ファンクションキー3（左側下段）、ファンクションキー4（右側下段）

12 キーボード端末の場合、キーボードの「Z」「X」キーにそれぞれ対応しています（ただし、テキスト編集では無効です）。
フルキーボードの場合、このボタンに対応するキーはキーボードにはありません。

8.16.1. 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（8.16 – ⑥ 参照）。

図 8-23 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
- 「マナーモード状態」
 - マナーモード OFF
 - マナーモード ON
- 「自動起動可否状態」 （i アプリ起動中は設定不可）

- ADF 指定 (ADF の LaunchAt キーによる個別自動起動可否設定)
 - 可
 - 不可
- API 指定 (自動起動時刻設定 API による個別自動起動可否設定)
 - 可
 - 不可

※アプリから自動起動可否設定を変更すると、その結果が反映されます

- 「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」
 - 内部電源
 - 外部電源
- 「バッテリー残量」
 - 1～100
- 「残り充電時間」
 - 数値入力後「適用」ボタンで反映
- 「残り利用時間」
 - 数値入力後「適用」ボタンで反映
- 「エリア状態」
 - 圏内 (HSDPA+HSUPA)
 - 圏内 (HSDPA)
 - 圏内 (FOMA)
 - 圏内 (ローミングアウト)
 - 圏外
 - セルフモード
- 「電波状態」
 - アンテナ 1 本
 - アンテナ 2 本
 - アンテナ 3 本
- 「UIM ロック状態」 (i アプリ起動中は設定不可)
 - 通常 (VERSION 1)
 - 通常 (VERSION 2)
 - 制限中
- 「歩数計状態」
 - 停止中
 - 計測中

※アプリから歩数計の状態を変更すると、その結果が反映されます

- 「身体活動量計状態」
 - OFF
 - ON

※アプリから身体活動量計の状態を変更すると、その結果が反映されます

- 「脈拍計測定状態」
 - 正常
 - 暗い

- 明るい
- 位置不正
- 未検出
- 「加速度センサ状態」
 - デバイス 1
 - OFF
 - ON

※加速度センサの「デバイス 0」は常に ON であり端末状態ウィンドウから状態を変更することはできません

- 「端末音声出力先状態」
 - 内部出力
 - 外部出力
- 「音楽再生状況表示」

音楽再生状況表示画面を表示します。

音楽再生状況表示画面については、「8.16.5 音楽再生状況表示」で詳しく説明します。

- ・ UIM ロック状態が制限中の場合は、エリア状態が圏外になります。
- ・ エリア状態が圏外の場合は、電波状態は無効になります。
- ・ i アプリ起動中に UIM ロック状態は変更できません。

8.16.2. オプションキーウィンドウの操作

オプションウィンドウは、エミュレーション端末のオプションキーウィンドウ表示を押すと表示されます（8.16 – ⑦ 参照）。



図 8-24 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 キー
 - 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
 - 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
- 「フルキーボードキー」
 - “ ” キー
 - “ . ” キー
 - “ ” キー
 - SHIFT キー（SHIFT キーを単独で押された状態）

※ 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。

8.16.3. 開発支援情報

開発支援情報画面は、エミュレーション端末の開発支援情報を押すと表示されます（8.16 – ⑧ 参照）。

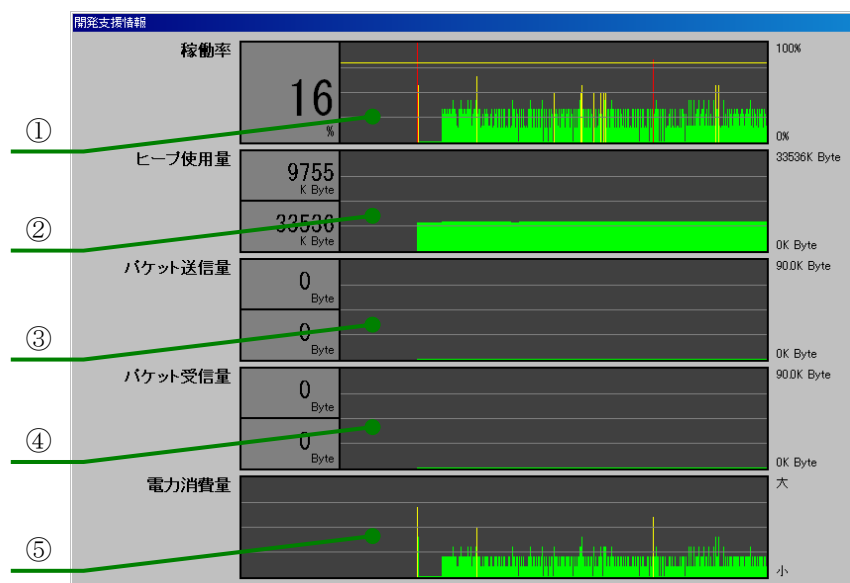


図 8-25 開発支援情報

開発支援情報画面では、現在起動中の i アプリ（フルアプリ）に関する各種情報を時系列でグラフ表示します。

① 稼働率

i アプリ動作中のシステムの稼働率を表示します。左端に現在の稼働率を、右側に時間経過グラフを表示します。

稼働率が閾値を超えると、そのときに実行している Java スレッドのスタックトレースをコンソールへ出力させることが可能です（初期設定状態ではスタックトレース出力は行いません）。

なお、稼働率はエミュレータが管理する「アイドル時間」の長さをもとに算出しているため、動作環境のスペックや、同時に起動中の他の Windows アプリケーションの稼働状況に影響されます。また、「スタックトレース出力」が ON の設定で i アプリを実行した場合、Java の JIT は使用しません。

② ヒープ使用量

i アプリ動作中のJavaヒープ使用量を表示します。左端に現在使用中のヒープサイズと使用可能な総ヒープサイズを、右側には総ヒープサイズを 100%とした場合のヒープ使用率の時間経過グラフを表示します。

なお、ここで使用する「総ヒープサイズ」は、アプリケーション動作環境設定画面で設定された「ヒープサイズ」（「6.2 全般設定」参照）から、システムで使用する（Javaから使用できない）領域を差し引いたものとなるため、一般にはアプリケーション動作環境設定画面で設定された「ヒープサイズ」よりも小さくなります。

③ パケット送信量

④ パケット受信量

i アプリ動作中のパケット送受信量を表示します。左端に現在のパケット送受信量とアプリ起動時からの累積送信量を、右側にはエミュレータ性能設定画面（「8.5 エミュレータ性能設定」参照）で設定された通信速度（未設定の場合は 7,200,000bps）に対する通信帯域使用率（理論値）の時間経過グラフを表示します。なお、HTTP通信の場合、表示されるパケット送受信量はHTTPボディ部のみのサイズであり、実際の i モード通信網による通信量とは異なります。

⑤ 電力消費量

i アプリ動作中の電力使用量の時間経過グラフを表示します。電力使用量は稼働率とパケット送受信量から算出しています。

なお、表示内容はアプリ起動時にクリアされます。またミニアプリ起動中は使用できません。

8.16.4. 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（8.16 – 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（「6.4. 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 8-26 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。

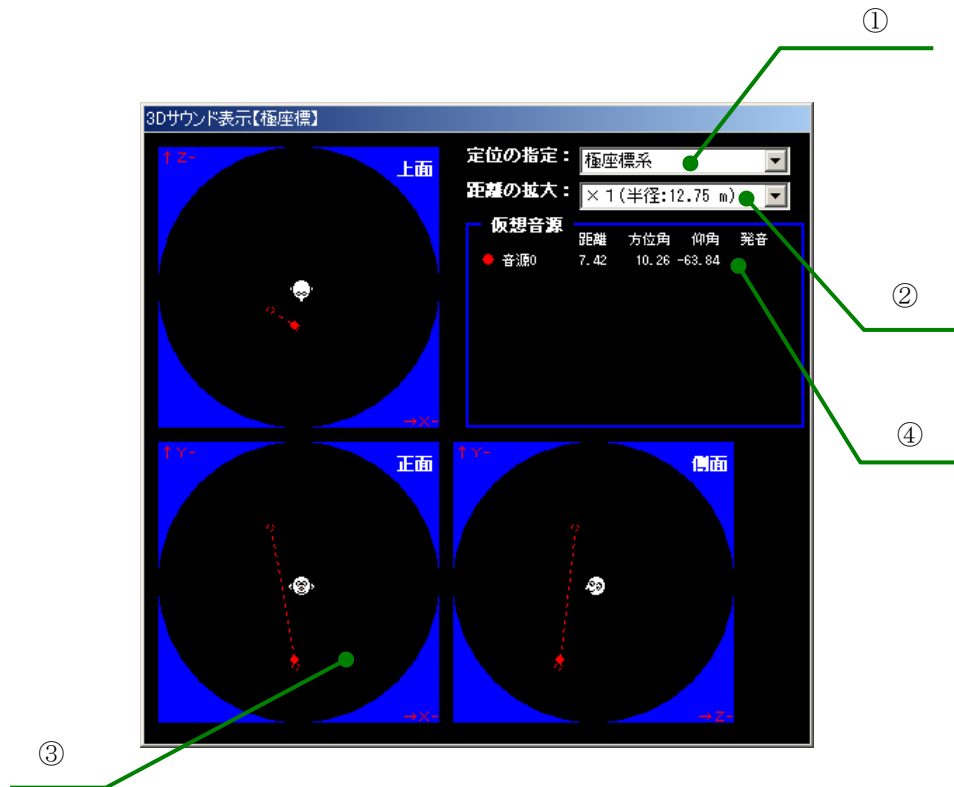


図 8-26 3D サウンド表示

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
「極座標系」か「デカルト座標系」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)。

8.16.5. 音楽再生状況表示

音楽再生状況表示画面は、端末状態ウィンドウの音楽再生状況表示ボタンを押すと表示されます (図 8-23 参照)。

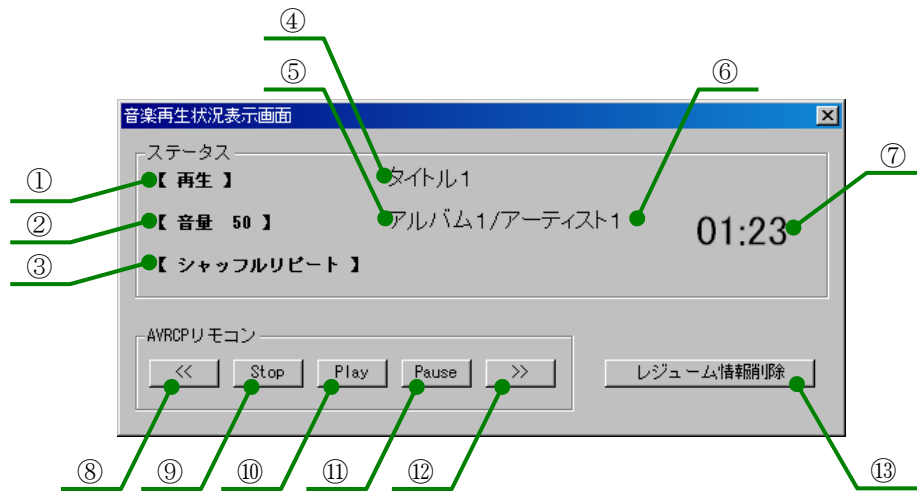


図 8-27 音楽再生状況表示画面

音楽再生状況表示画面では、音楽プレイヤー機能 API から操作する音楽再生ソフトウェアのエミュレーション状況を表示します。

また、Bluetooth AVRCP リモコンからの操作イベントを発生します。

[ステータス]

現在の音楽再生ソフトウェアの状態を表示します。

- ① 現在の音楽再生ソフトウェアの状態
- ② 現在の音量
- ③ 現在の音楽再生ソフトウェアの再生モード
- ④ 現在再生中の音楽コンテンツのタイトル情報
- ⑤ 現在再生中の音楽コンテンツのアルバム情報
- ⑥ 現在再生中の音楽コンテンツのアーティスト情報
- ⑦ 現在再生中の音楽コンテンツのタイムカウンタ

[AVRCP リモコン]

AVRCP リモコンからのオペレーションに伴うイベントをエミュレーションします。

StarApplication#addEventListener にて StarEventObject.

STAR_BLUETOOTH_OPERATION_RECEIVED を指定して StarEventListener が登録されている場合は、リスナに BluetoothEvent が通知されます。また、サスペンド中

(StarApplication#suspend()によるサスペンドを除く)は音楽再生ソフトウェアへボタン操作が直接通知され、操作内容に応じて音楽再生ソフトウェアの状態が変化します。

- ⑧ 1 曲戻し要求 (OPERATION_AVRCP_BACKWARD)
- ⑨ 停止要求 (OPERATION_AVRCP_STOP)
- ⑩ 再生要求 (OPERATION_AVRCP_PLAY)
- ⑪ 一時停止要求 (OPERATION_AVRCP_PAUSE)
- ⑫ 1 曲送り要求 (OPERATION_AVRCP_FORWARD)

[レジューム情報削除]

- ⑬ レジューム情報削除

音楽再生ソフトウェアで保持するレジューム情報 (前回終了時の状態) を強制的にクリアします。

第9章 ツール

この章では、「iα ppliTool」で扱える便利なツールについて説明します。

9.1. 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。

ミニアプリにおけるスクラッチパッドは、最初の 1 件目です。

9.1.1. スクラッチパッドファイル

スクラッチパッドは、実際の携帯電話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、通常の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:¥iDKStar2.0¥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（「4.8. 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。

9.1.2. 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 9-1）。

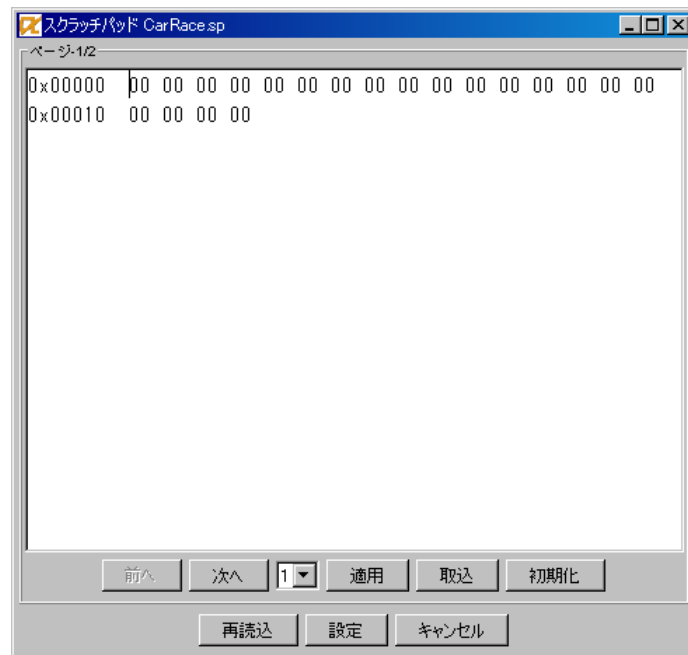
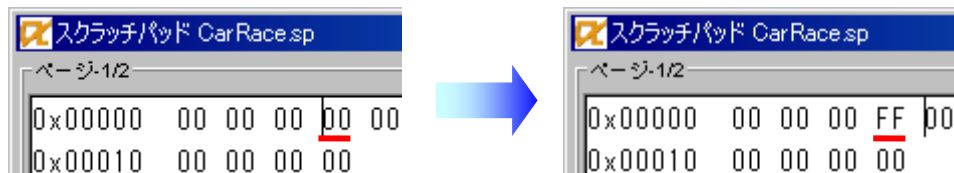


図 9-1 スクラッチパッド管理ツール

2. 現在、読み込まれているプロジェクトの最初のスク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つ前のスク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つ先のスクラッチパッドの内容が表示されます。
6. プルダウンからスクラッチパッドの連番を選択すると、そのスクラッチパッドの内容が表示されます（図 9-2）。



図 9-2 スク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 9-3）。

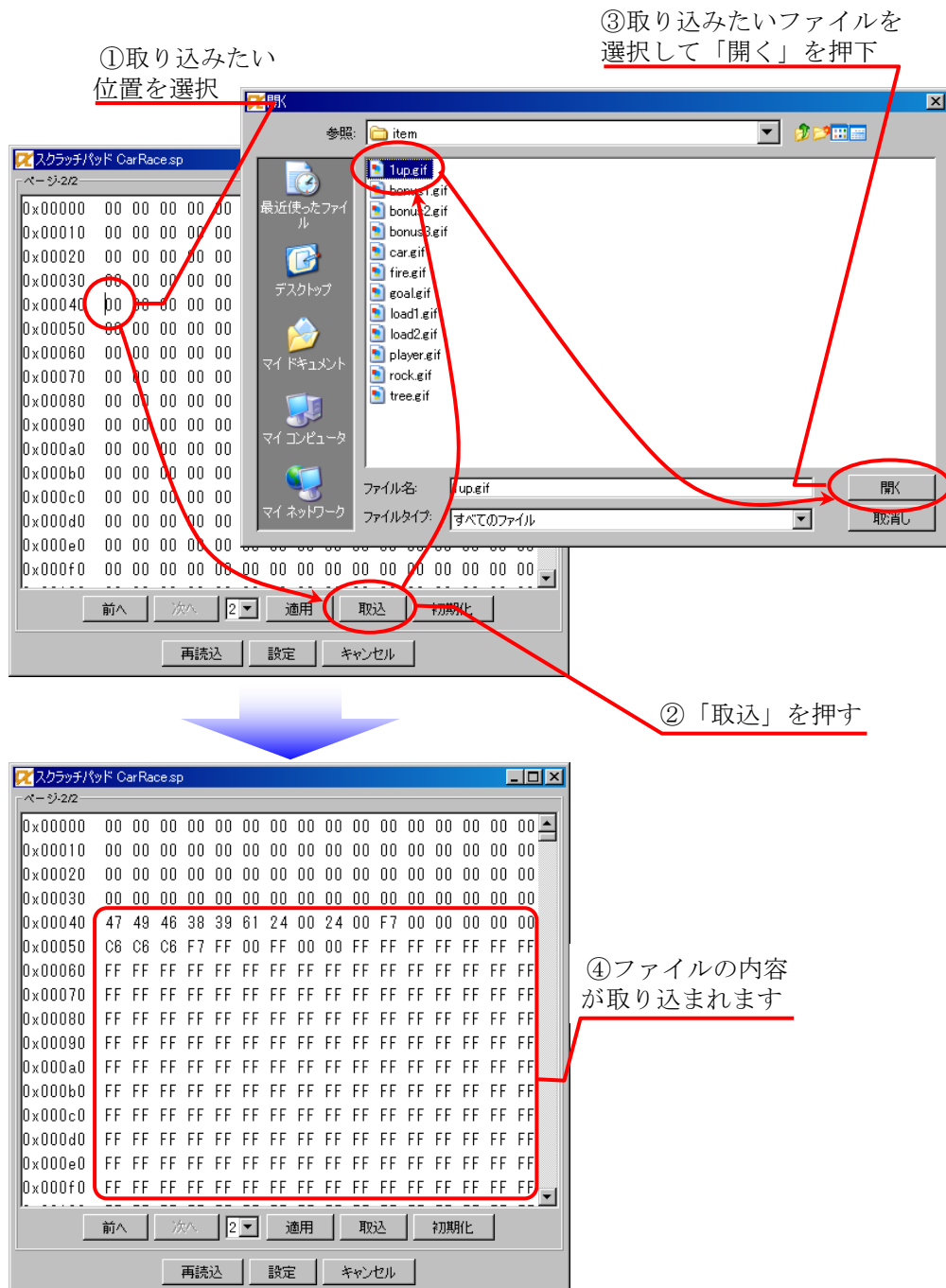


図 9-3 スクラッチパッドの置換処理

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め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 9-4）。

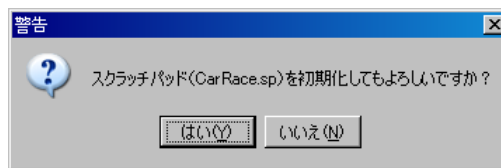


図 9-4 スクラッチパッド削除確認ダイアログ

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み込みし、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。

9.1.3. 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ファイル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（「4.8. 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。

9.2. 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」→「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 9-5）。

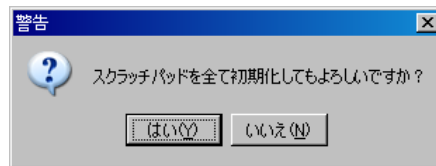


図 9-5 スクラッチパッド削除確認ダイアログ（全て）

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。

9.3. 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 データファイルの内容を表示・編集するためのツールです。

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事ができます。

9.3.1. 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されます：

C:\¥iDKStar2.0¥apps¥<プロジェクト名>¥<プロジェクト名>.fc

9.3.2. 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 9-6）。

インストール直後のデータファイルにはデータ領域を除き適切な値が設定されています。仮想FeliCaデータファイルの構造については、「15. 29. 仮想FeliCaデータファイル」を参照してください。

「表示切替」ボタンを押す事で、階層表示～バイナリ表示の切替が可能です。

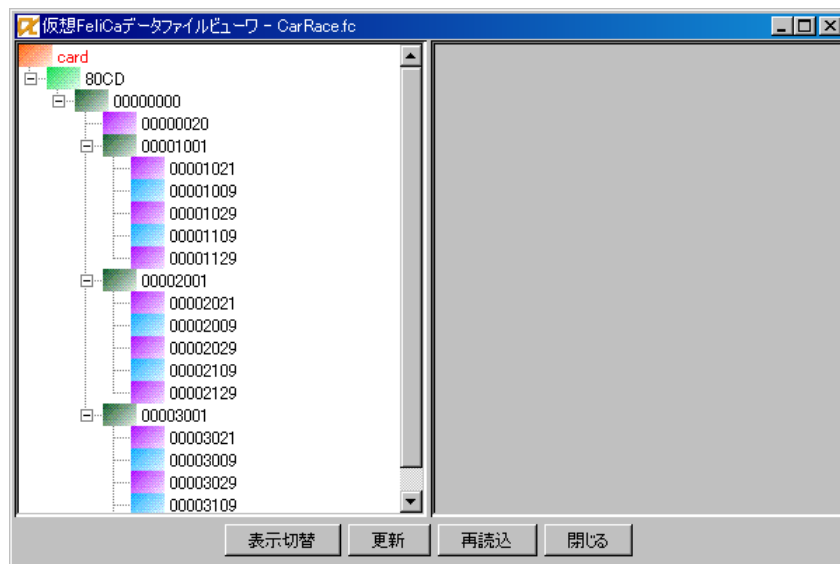


図 9-6 仮想 FeliCa データファイルビューワ

9.3.3. 階層表示

階層表示では、仮想FeliCaデータのブロックの関係をツリー構造で表示します（図 9-7）。

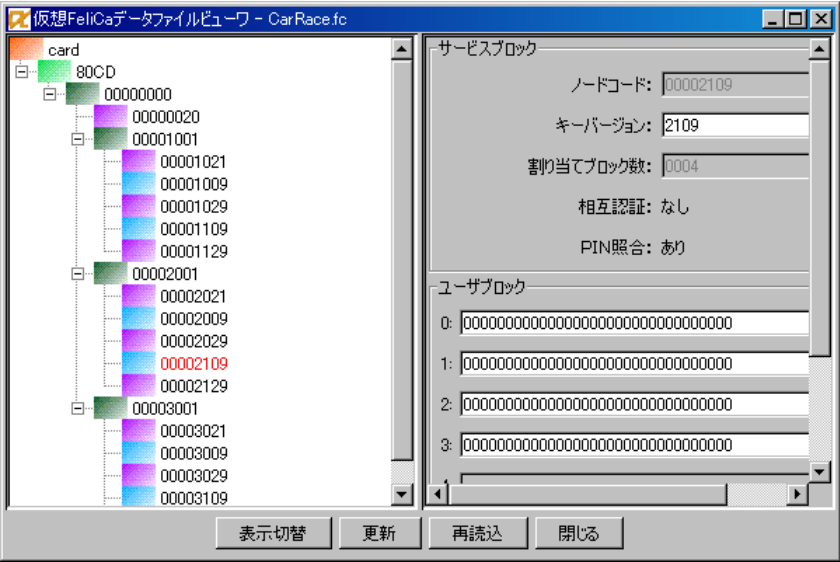


図 9-7 仮想 FeliCa データファイルビューワ（階層表示）

(1) ノードの表示

各ノードを示すアイコン及び表示色は表 9-1 のようになります。

表 9-1 階層表示におけるノード表示

アイコン	表示色	ノード
	橙	カード
	黄緑	システム
	緑	エリア
	水色	サービス
	紫	PIN サービス

(2) ノードのブロック情報表示

ノードコードツリーで選択されているノードに関連するブロック情報を表示します。
選択中のノードによって、表示される情報は表 9-2 のようになります。

表 9-2 選択ノードのブロック情報表示項目

選択中のノード	表示項目	編集
カード	—	—
システム	システムコード キーバージョン	× ○
エリア	ノードコード キーバージョン エンドサービスコード 子エリア（の作成可・不可） PIN 照合（の有無）	× ○ × × ×
サービス	ノードコード キーバージョン 割当ブロック数 相互認証（の有無） PIN 照合（の有無） ユーザブロック	× ○ × × × ○
PIN サービス	ノードコード キーバージョン PIN 照合試行回数 照合試行上限回数 PIN 属性（B8） PIN 属性（Bc）	× ○ ○ × × × ×

(3) ノード情報の編集

ノード情報を編集するには：

1. ノードコードツリーから任意のノードを選択すると、ビューワ右部に選択ノードのブロック情報が表示されます。
2. ブロック情報の任意の内容を変更し、ノードコードツリーの他のノードを選択するか、「更新」ボタンを押すと確定されます。

- ※ テキストフィールドには値は全て 16 進数で入力してください。
- ※ 各項目を変更し、ノードコードツリーの他のノードを選択して編集を確定した時点では、仮想 FeliCa データファイルには反映されていません。仮想 FeliCa データファイルへの反映は「更新」ボタンを押してください。
- ※ 階層表示では、キーバージョン、ユーザブロック、PIN を編集することができます（表 9-2）。
- ※ 「再読込」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。
- ※ 「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。

9.3.4. バイナリ表示

バイナリ表示では、仮想FeliCaデータのブロックをバイナリ形式で表示します（図 9-8）。



図 9-8 仮想 FeliCa データファイルビューワ（バイナリ表示）

(1) ブロックの表示

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

(2) 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

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

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。

(3) ブロックの編集

ブロックを編集するには：

1. 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。
該当する記憶領域に下線が移動します。
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。
2. 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。
3. 「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。
 4. 「再読込」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。
 5. 「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。

9.3.5. フリー領域を扱うiアプリの利用

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（「4.8. 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（「4.8. 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（「4.8.2. FeliCaフリー領域の関連付け」参考）。

階層表示で手動設定するには：

1. ノードコードツリーから サービスノード 0000*009 を選択します。
*は 1～3 の任意のノードコード
2. CPID サービスのブロックの情報が表示されるので、ユーザブロックを ADF に設定した FeliCaCPID と同じ値に変更します。
3. ノードコードツリーから PIN サービスノード 0000*129 を選択します。
*は 1～3 の任意のノードコードで、手順 1 と同じ値を指定します。

4. PIN サービスのブロックの情報が表示されるので、PIN を ADF に設定した FeliCaPIN の値に変更します。例えば、ADF に設定した FeliCaPIN の値が 12345678 の場合は、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。

バイナリ表示で手動設定するには：

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。
- 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。

9.4. 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、「9.7. 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 9-9）。

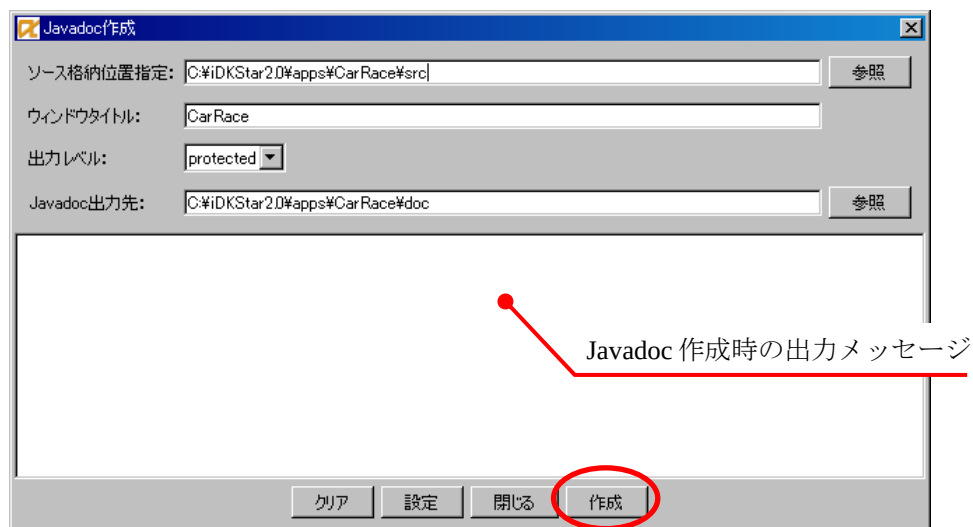


図 9-9 javadoc 作成ツール

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 9-3 に示すとおりです。

表 9-3 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、表 9-4 に示すとおりです：

表 9-4 javadoc.properties ファイルの記述方法

項目	内容
srcdir = C:¥iDKStar2.0¥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:¥iDKStar2.0¥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 • linkoffline • excludedocfilessubdir • serialwarn • taglet • tagletpath • 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 jdk1.5.0_19 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いません。（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。

9.5. ADF チェックツール

ADF チェックツールは、HTTP サーバ上に配置された i アプリの ADF (.jam ファイル) 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、ADF が間違っても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場合があります。

本ツールでは、こうした現象を解消するため、HTTP サーバ上に設置された i アプリの ADF を読み取ってチェックする事ができます。

ADF チェックツールを起動するには：

1. メニューから「ツール」→「ADFチェック...」を選択します（図 9-10）。

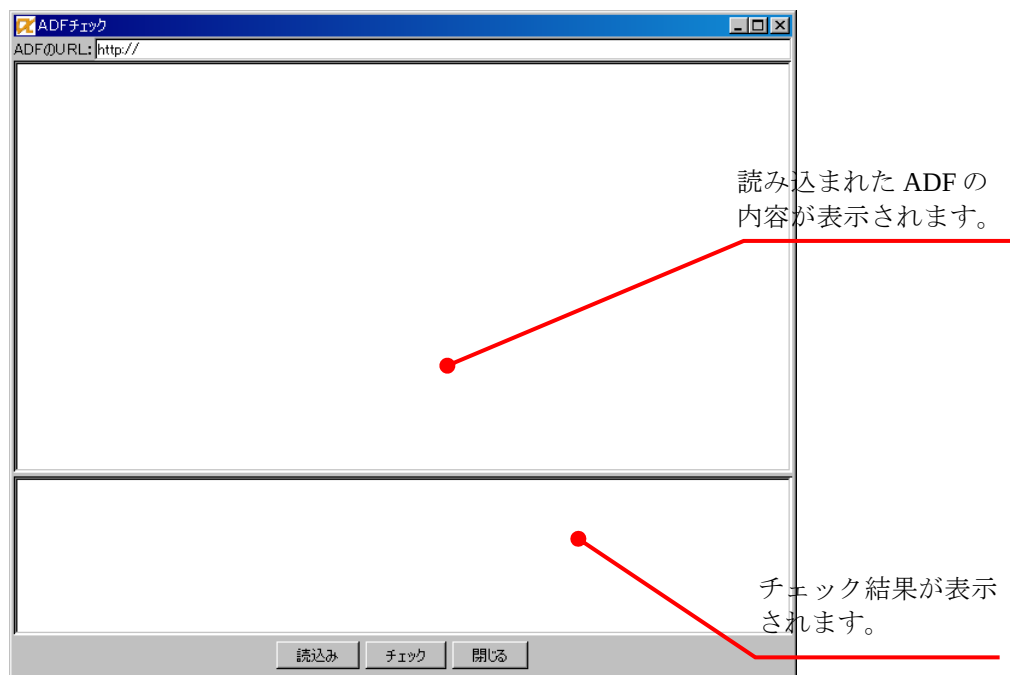


図 9-10 ADF チェックツール

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:/iDKStar2.0/apps/CarRace/bin/CarRace.jam)
3. 「読み込み」ボタンを押すと ADF を HTTP 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 9-11）。

「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。

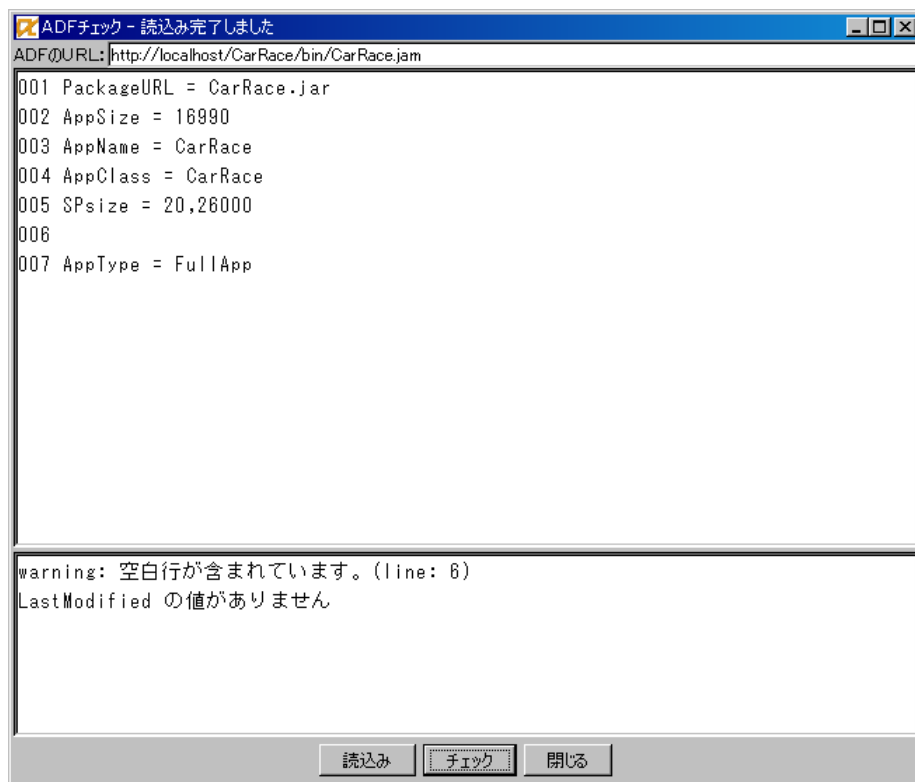


図 9-11 ADF のチェック実行

- ※ 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 ファイル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 が正しいかどうか確認してみてください。

ワンポイント

ADF には i アプリ (JAR ファイル) 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 ファイル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 ファイルのみの置き換えはできないものと考えましょう。)

9.6. 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 9-12）。
（あらかじめ、ショートカットキーを設定しておく则便利です（「8.3. ショートカットキーの設定」参照）。）

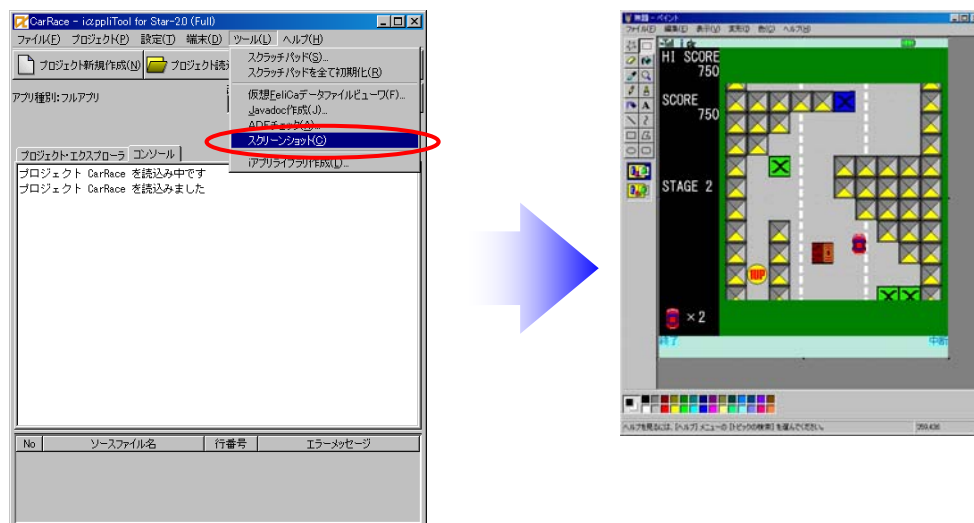


図 9-12 スクリーンショットの実行

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、「8.8. スクリーンショット保存の設定」を参考にしてください。

9.7. 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。

9.7.1. 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 9-13）。

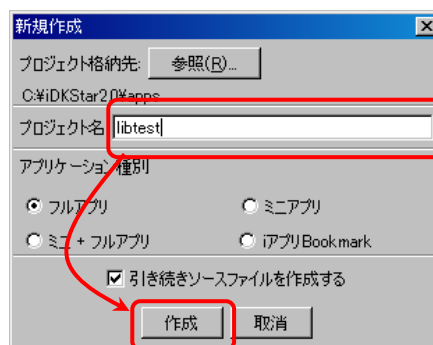


図 9-13 i アプリライブラリ プロジェクト新規作成

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 9-14）。



図 9-14 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 9-15)。

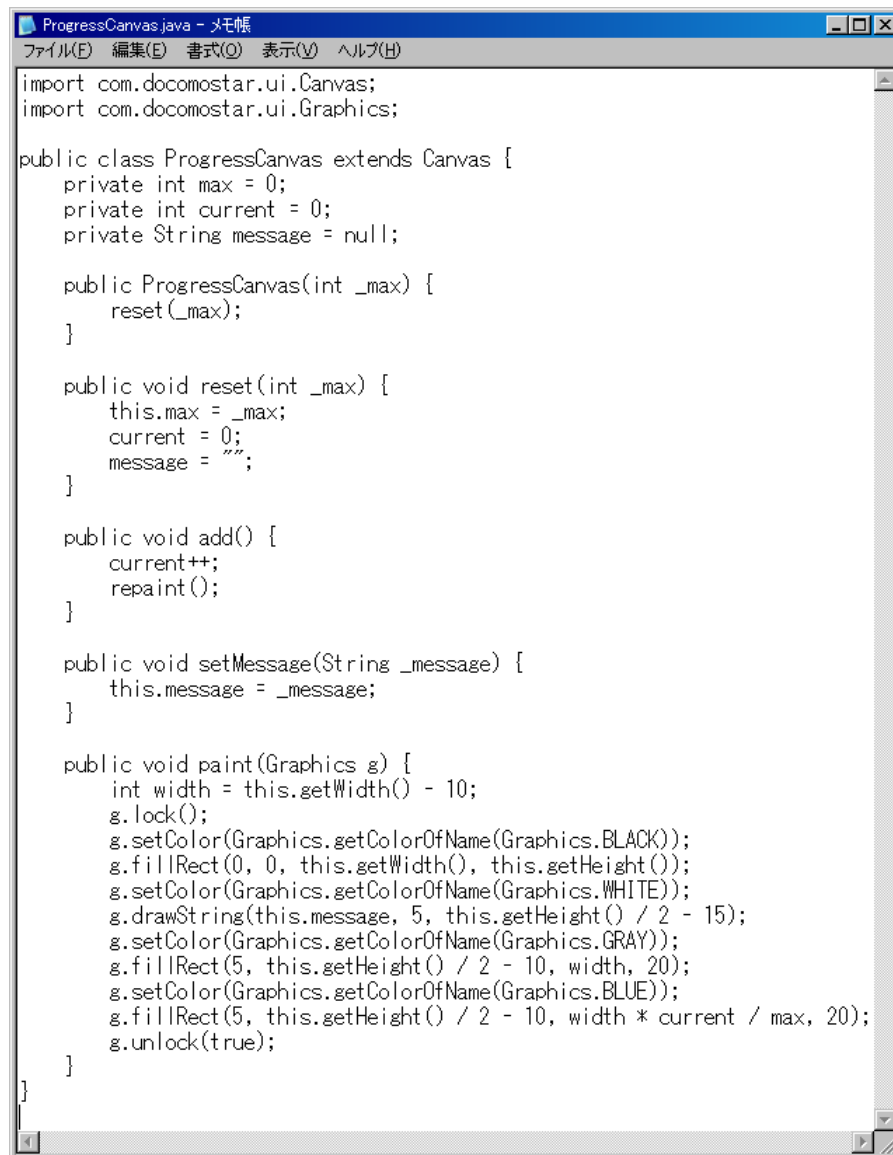


図 9-15 i アプリライブラリ プログラムの作成

(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 9-16)。

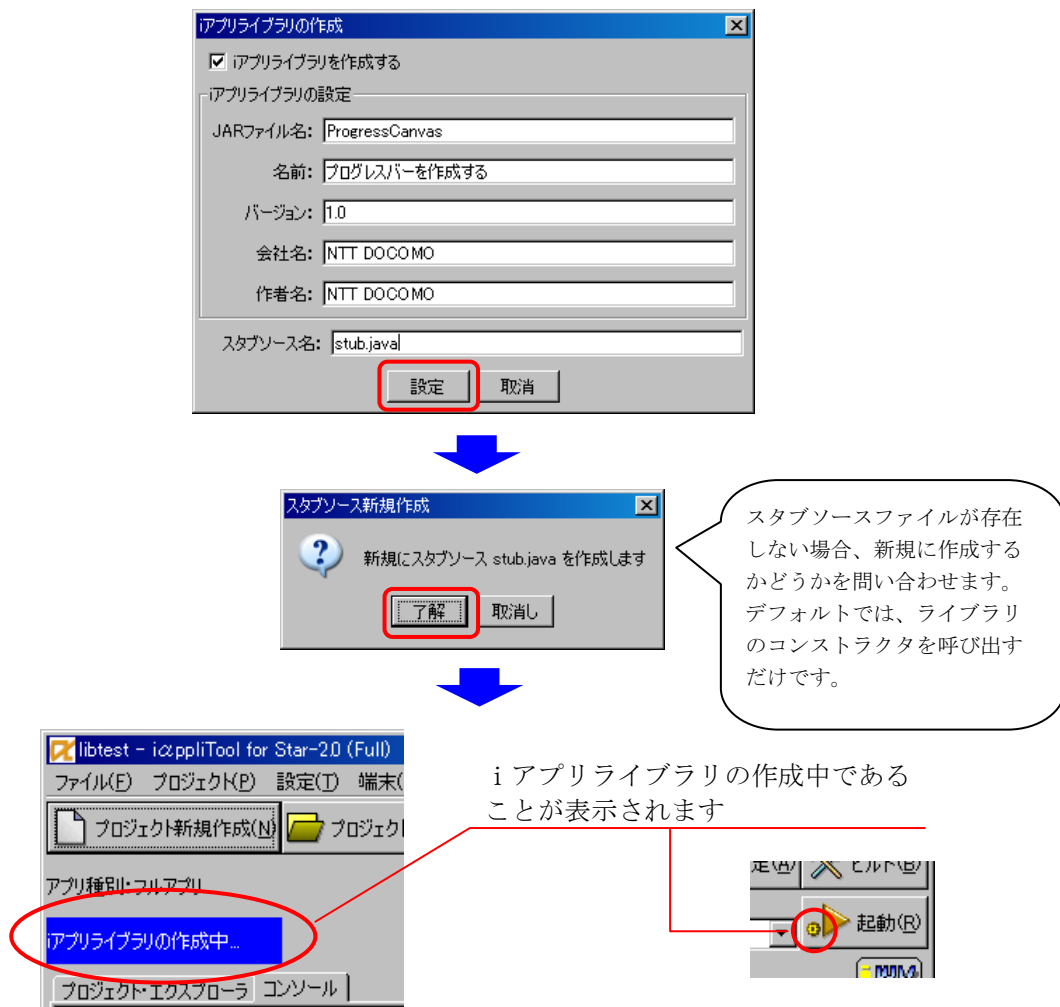


図 9-16 i アプリライブラリの作成

ここで設定した属性は、i アプリライブラリ中の JAR ファイルに 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 9-5 に示すとおりです。

表 9-5 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 ^[12] から自動的に作成されます (図 9-17)。 省略すると“プロジェクト名_stub.java”が作成されます。	xxx_stub.java

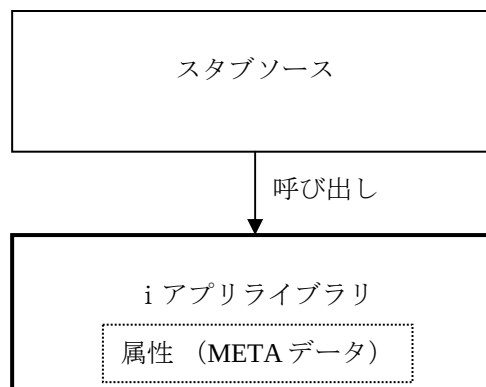


図 9-17 スタブソースと i アプリライブラリの関係

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 9-18)。

^[12] 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。

C:\¥iDKStar2.0¥lib¥ComponentStub.txt

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。

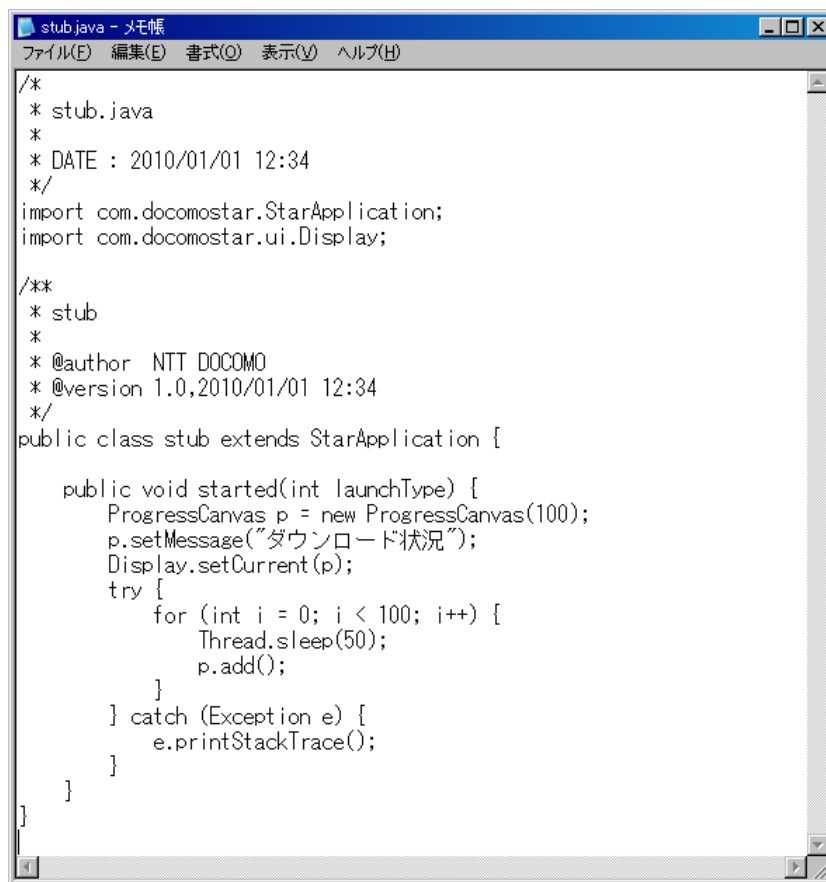


図 9-18 i アプリライブラリ スタブソースの書き換え

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。

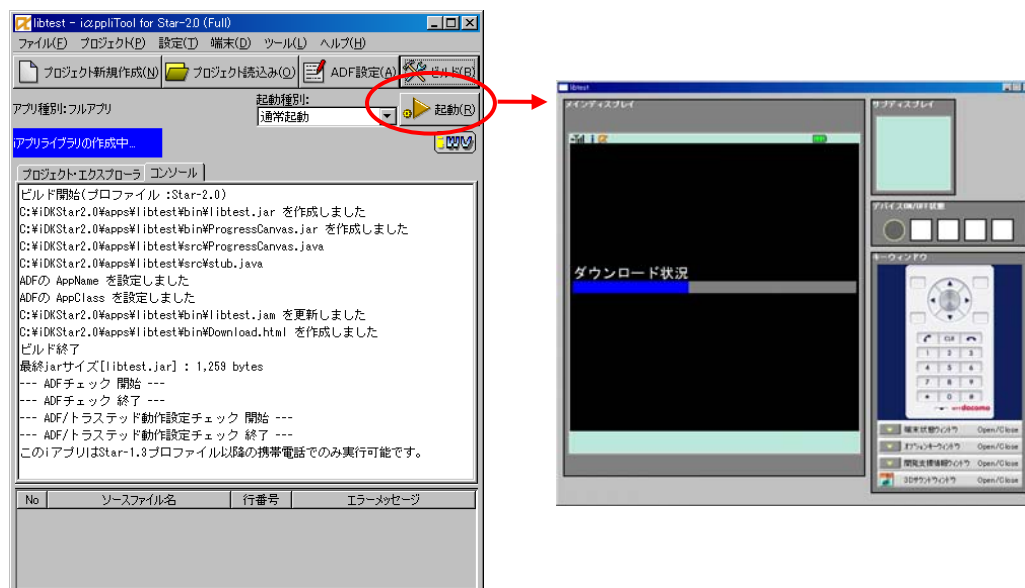


図 9-19 i アプリライブラリの実行

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:¥iDKStar2.0¥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。

9.7.2. 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。

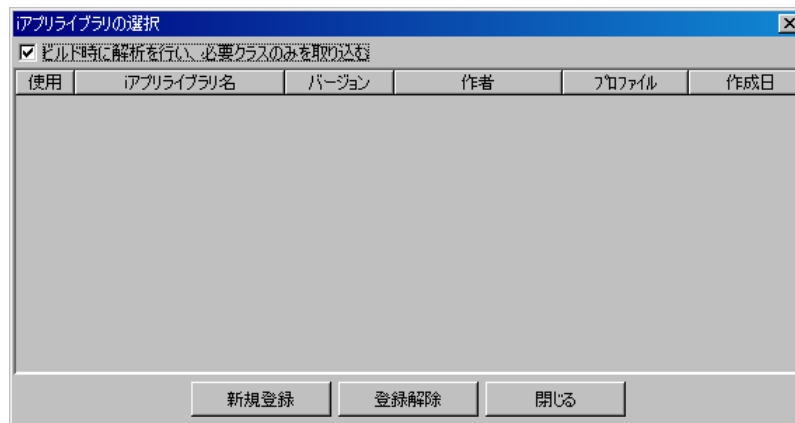


図 9-20 i アプリライブラリの選択画面

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 ファイル) 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 9-21 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

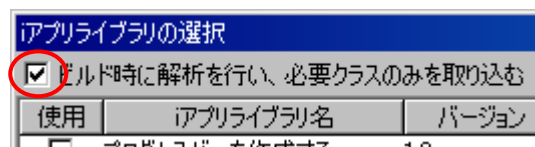
次の場所に i アプリライブラリを置いておくと自動的に登録できます。

C:\¥iDKStar2.0¥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 ファイル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:\¥iDKStar2.0¥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。

必要なものだけにチェック

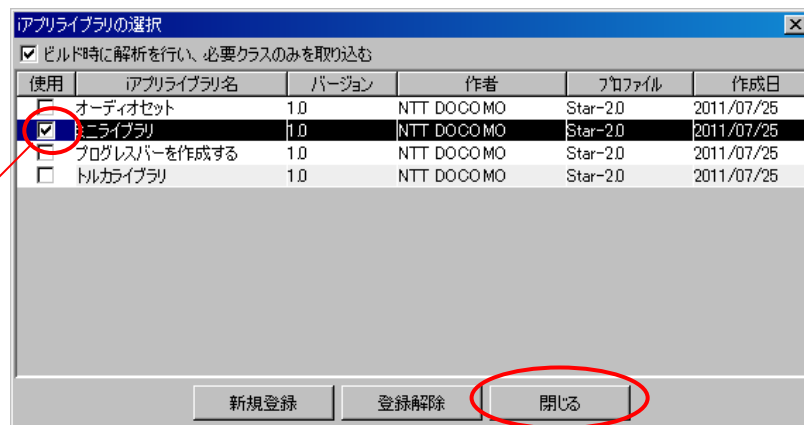


図 9-22 ライブラリの選択

8. 「閉じる」を押します。
9. i アプリをビルドします。
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。

9.8. 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、「4.8. 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 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。

これらのうち、ドコモがユーザ証明書を発行する電子認証サービス「FirstPass」は 2012 年 8 月 31 日をもってサービスを終了しました。そのため 2012 年 9 月 1 日以降、9.8.1 項で解説する、ドコモが発行したユーザ証明書を使用する方法で署名をしても i アプリで利用することはできません。

なお、9.8.2 項で解説する、ユーザが独自に発行した証明書を使用する方法は、2012 年 9 月以降も利用可能です。

9.8.1. 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 携帯電話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** で携帯電話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 9-23 に示します。

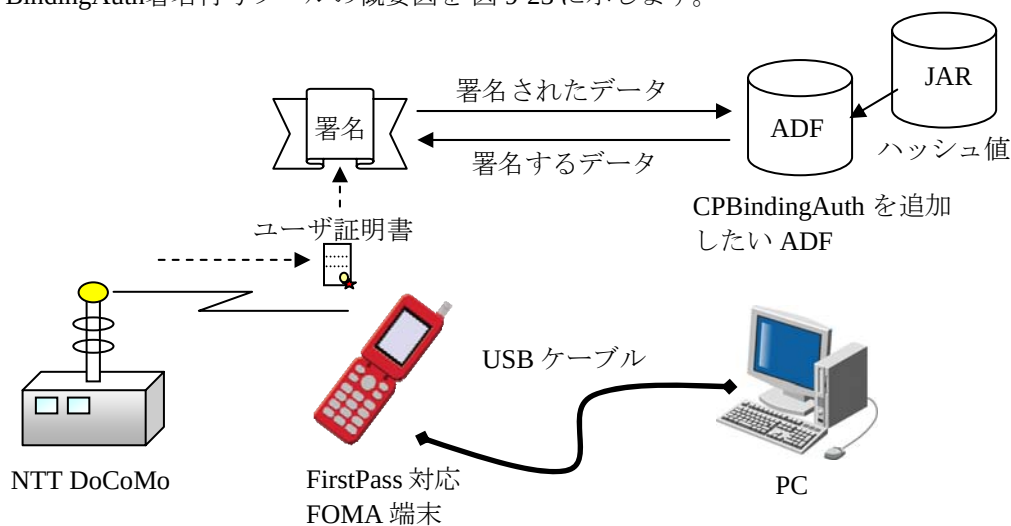


図 9-23 CPBindingAuth 署名付与ツール 概念図

CPBindingAuth を設定するには：

1. FOMA 携帯電話で 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 携帯電話と 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:¥iDKStar2.0¥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 ADF  
              (UseStorage が設定されている必要があります。)  
JAR          : i アプリの JAR ファイル  
PIN2_CODE    : FOMA 携帯電話の PIN2 コード  
COM_PORT     : USB ケーブルを接続している COM 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 に CPBindingAuth と PackageHashCode が追加されています。
このとき既に ADF に CPBindingAuth や PackageHashCode が存在する場合は書き換えられます。

9.8.2. 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、ADF に CPBindingAuth 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:¥iDKStar2.0¥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 ADF  
              (UseStorage が設定されている必要があります。)  
JAR          : i アプリの JAR ファイル  
PKCS#12      : PKCS#12 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 に CPBindingAuth と PackageHashCode が追加されています。
このとき既に ADF に CPBindingAuth や PackageHashCode が存在する場合は書き換えられます。

第10章 iαppliTool の終了

「iαppliTool」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。

「iαppliTool」を終了しても「iαppliTool」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11章 デバッグトレース

高度な使い方として、コンソールに VM のデバッグトレースを出力する事ができます。
この機能は、フルアプリのみ実行できます。

デバッグトレースを実行するには：

1. 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 11-1）。

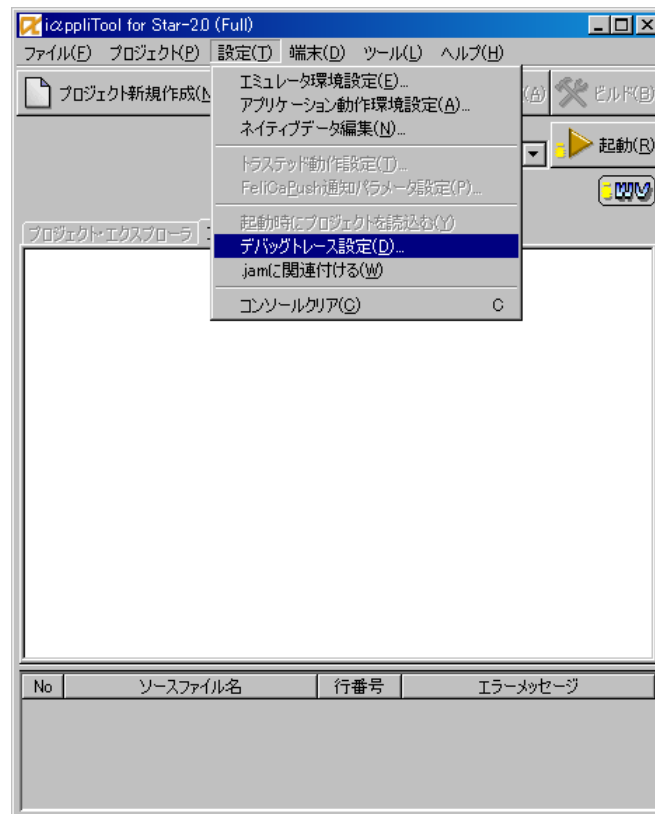


図 11-1 デバッグトレースの設定

2. デバッグトレース設定画面（図 11-2）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】

-verbose:class.....trace class loading
-verbose:gc.....trace garbage collection

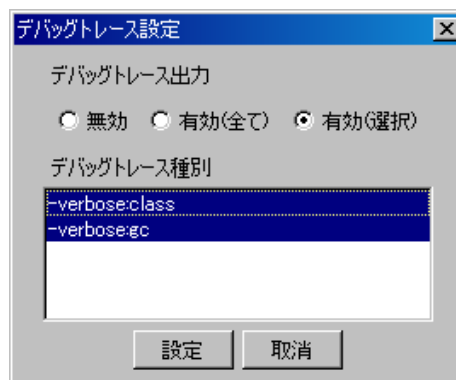


図 11-2 デバッグトレース設定画面

3. トレースしたいプロジェクトを読み込んで、アプリを起動します。
この時、i アプリはデバッグ用の VM 環境で実行されます。
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12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携帯電話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、`Launcher.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携帯電話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、`Launcher.launch(LAUNCH_STARAPPLI)` で起動できます。

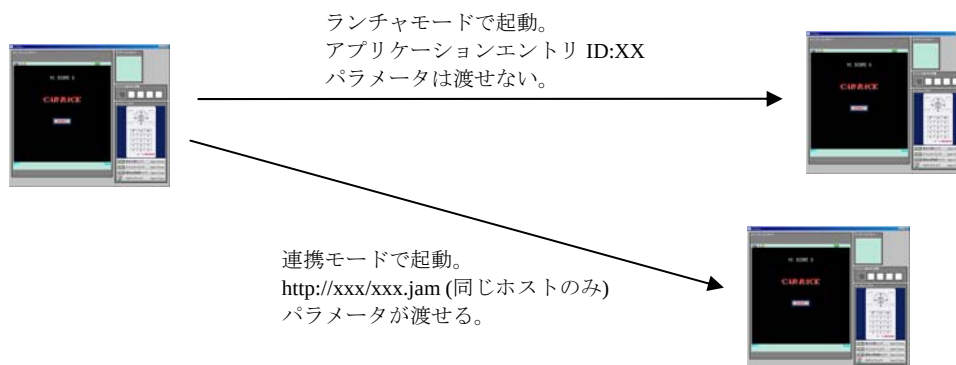


図 12-1 i アプリ連携の実行

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ミニアプリでないこと。
- 起動先アプリケーションがミニアプリでないこと。
- 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」（「4.9 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（「4.8. 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:¥iDKStar2.0¥apps」）に作成されている必要はありません。

第13章 フルミニ連携

「フルミニ連携」とは、フル+ミニアプリまたはミニ+フルアプリにおいて、同一パッケージのミニアプリからフルアプリ、フルアプリからミニアプリに切り替える機能です（図 13-1）。

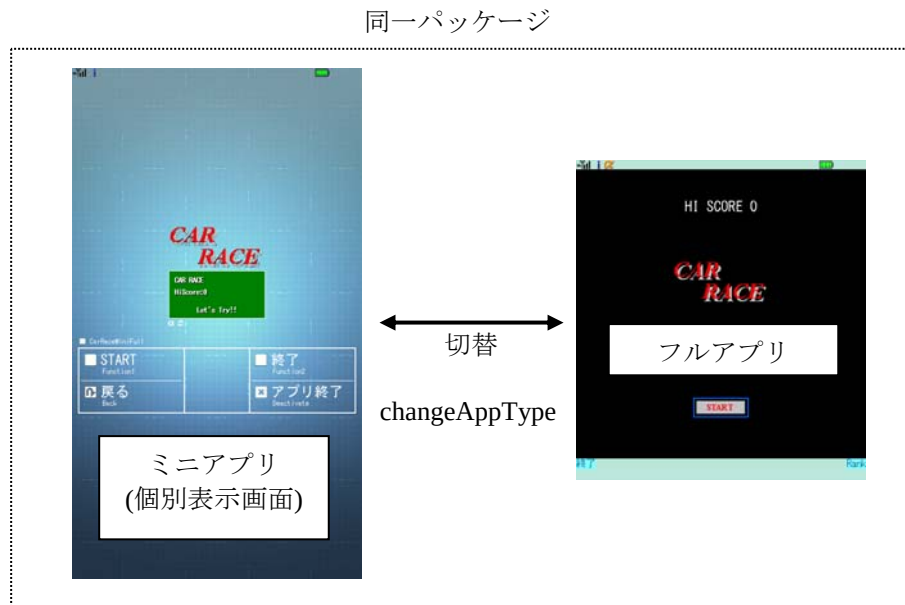


図 13-1 フルミニ連携

同一パッケージ（フル+ミニ）における連携は、ADF の LaunchApp キーに関係なく、StarApplication#changeAppType()で切り替えられます。

同一パッケージにおいて、フルアプリからミニアプリを起動した場合の起動タイプは、「LAUNCHED_FROM_FULLAPPLI」です。

同一パッケージにおいて、ミニアプリからフルアプリを起動した場合の起動タイプは、「LAUNCHED_FROM_MINIAPPLI」です。

第14章 ステレオ表示機能

エミュレータでは、i アプリオプション API のステレオ表示機能利用時に、実際に画面上でステレオ表示をすることができます。

本章では、エミュレータでのステレオ表示方式および手順について説明します。

i アプリオプション API のステレオ表示機能の仕様、および対応 i アプリの作成方法につきましては、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。

14.1. 表示方式

エミュレータでは、以下の2通りの方法でステレオ表示することができます。

- サイドバイサイド表示
左目用画像と右目用画像を左右にそれぞれ表示し、平行法あるいは交差法を用いて裸眼で立体視する方式です。
立体視するにはある程度慣れが必要ですが、通常のモニターで表示可能です。
 - 平行法
左側に左目用画像を、右側に右目用画像を配置し、それぞれの側の目で平行に画像を見るようにする方式です。
視線は遠くを見るようなイメージで焦点を画面に合わせない状態とし、中央に見えてくる画像に意識を向けることで立体に見えるようになります。
 - 交差法
左側に右目用画像を、右側に左目用画像を配置し、視線を交差させてそれぞれ反対側の画像を見るようにする方式です。
画面との中間あたりに視線を合わせるようなイメージで視線を交差させて「より目」の状態とし、中央に見えてくる画像に意識を向けることで立体に見えるようになります。
- 3D表示
NVIDIA社が提供している、3D Visionテクノロジーに対応した環境で立体視する方式です。
専用のデバイスを利用することでより簡単に立体視を行うことが可能ですが、3D Vision- Readyの環境が必要になります。
3D Visionテクノロジー、および 3D Vision- Ready環境については、以下を参照ください。
<http://www.nvidia.co.jp/object/3d-vision-main-jp.html>

これらの2つの表示方式について、以下に表示手順を説明します。

14.2. サイドバイサイド表示

エミュレータでは、ステレオ表示を有効にした場合、デフォルトではサイドバイサイド表示方式（平行法）でステレオ表示されます（図 14-1）。

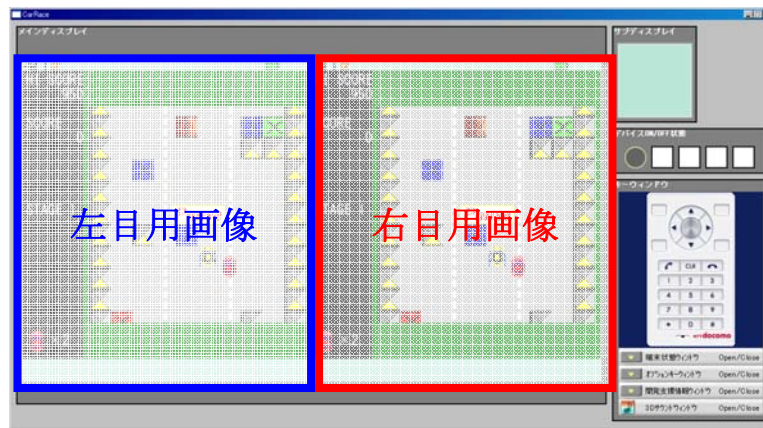


図 14-1 サイドバイサイド表示（平行法）

14.2.1. サイドバイサイド表示方式切り替え

サイドバイサイド表示方式（平行法、交差法）を切り替える場合は、LCD 上でマウスを右クリックすることで表示されるメニューを選択します。

サイドバイサイド表示（交差法）に切り替える場合は、メニューから「サイドバイサイド表示」の「交差法」を選択することで切り替わります（図 14-2）。

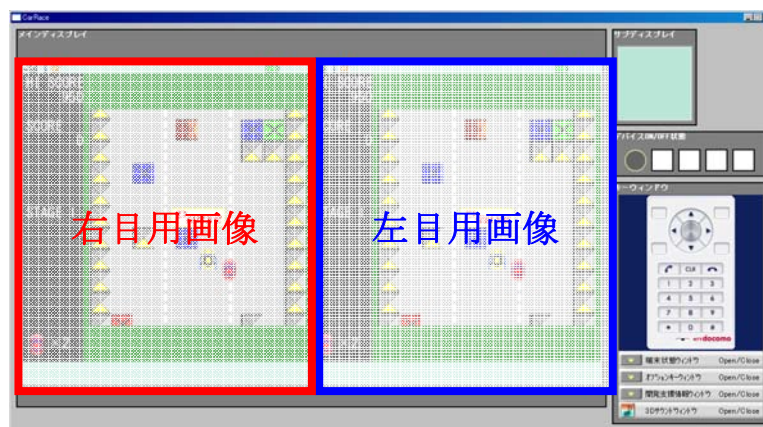


図 14-2 サイドバイサイド表示（交差法）

サイドバイサイド表示（平行法）に切り替える場合は、メニューから「サイドバイサイド表示」の「平行法」を選択してください。

なお、サイドバイサイド表示のメニュー項目は、ステレオ表示が有効になっている場合のみ選択可能です。

14.2.2. 解像度比設定による表示切り替え

サイドバイサイド表示時の i アプリ描画領域の表示サイズは、「8.6 エミュレーション端末ウィンドウの設定」における立体視の解像度比の設定、および解像度比が 2 の場合に設定可能な立体視拡大率切替により変わります。

「8.6 エミュレーション端末ウィンドウの設定」における立体視の解像度比設定で「解像度比: 1」を設定した場合は、描画領域サイズが横方向に 2 倍になり、画面を 2 分割してそれぞれ等倍のサイズで表示されます（図 14-3）。

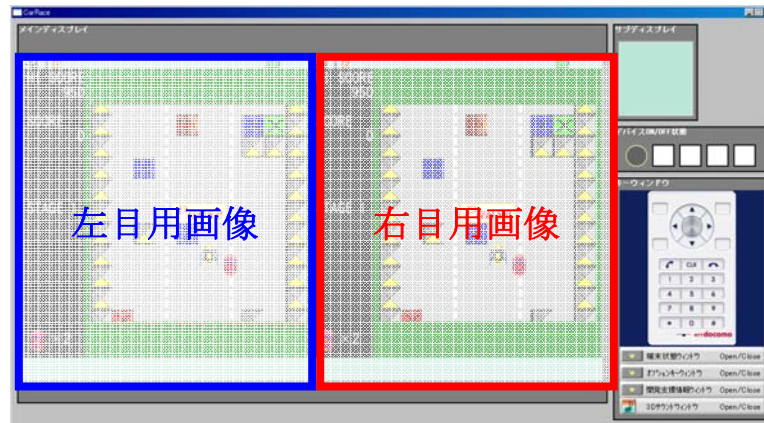


図 14-3 サイドバイサイド表示（解像度比: 1）

これに対し、「解像度比: 2」を設定した場合は、描画領域サイズはそのままとなり、画面を 2 分割してそれぞれ半分のサイズで表示されます（図 14-4）。

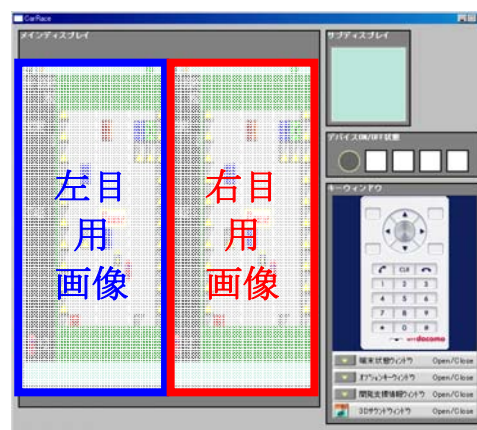


図 14-4 サイドバイサイド表示（解像度比: 2）

上記より、解像度比: 2 を選択した場合、元の i アプリ画面のアスペクト比を保たずに縮小表示されますが、エミュレータでは、利便性向上のため、LCD 上でマウスを右クリックすることで表示されるメニューから拡大表示ができます。

拡大表示したい場合は、メニューから「立体視拡大率切替」の「横 2 倍拡大表示」を選択します（図 14-5）。

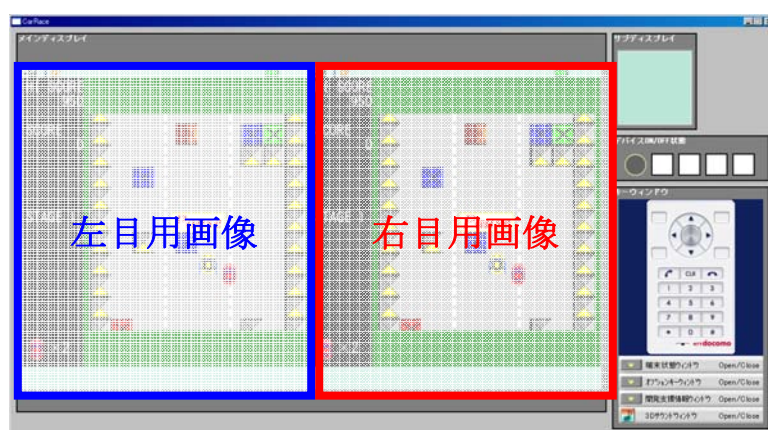


図 14-5 サイドバイサイド表示（解像度比: 2、横 2 倍拡大表示）

元に戻す場合は、メニューから「立体視拡大率切替」の「等倍表示」を選択してください。

14.3. 3D 表示

エミュレータでは、3D Vision に対応した環境（3D Vision-Ready の環境）において、3D Vision による立体視表示ができます。

但し、3D Vision に対応した全ての環境での動作を保証するものではありません。

3D 表示を行うには、3D Vision に対応した環境において、ステレオ表示を有効にした状態で、LCD 上でマウスを右クリックすることで表示されるメニューから「3D 表示」を選択してください。

3D 表示中は、常に全画面（フルスクリーン）表示となります。

F7 キーを押すと画面の等倍表示・フィットサイズ表示（モニターのサイズに合わせた拡大表示）の切り替えが可能です。

F8 キーを押すと 3D 表示が終了し、通常のエミュレータ画面に戻ります。

キーボード操作はエミュレーション端末ウィンドウと同様です。

なお、3D 表示中はマウス操作が無効となりますので、以下の操作は不可となります。

- LCD 上でマウスを右クリックすることで表示されるメニュー操作全般
- タッチデバイスの操作全般
- ポインティングデバイスの操作全般

また、「iα ppliTool」の操作は行えません。

これらの操作を行う場合は、F8 キーを押して 3D 表示を終了してください。

14.4. ステレオ表示のスタイル切替

ステレオ表示機能 API では、StereoScope#setStereoscopicStyle()を設定することで、ステレオ表示の表示方向を縦方向もしくは横方向に設定することができます。

エミュレータでは、この表示方向に合わせてステレオ表示を行います。

ステレオ表示中は、右クリックメニュー「ディスプレイ状態」の選択に伴うディスプレイ回転表示は行いません。ただし、横方向ステレオ表示時に、「ディスプレイ状態」の選択状態によって 90 度回転の方向は変化します。

縦方向を設定（StereoScope#setStereoscopicStyle()でSTEREOSCOPIC_STYLE_VERTICALを設定）した場合は、通常どおり縦方向でステレオ表示を行います。（図 14-6）

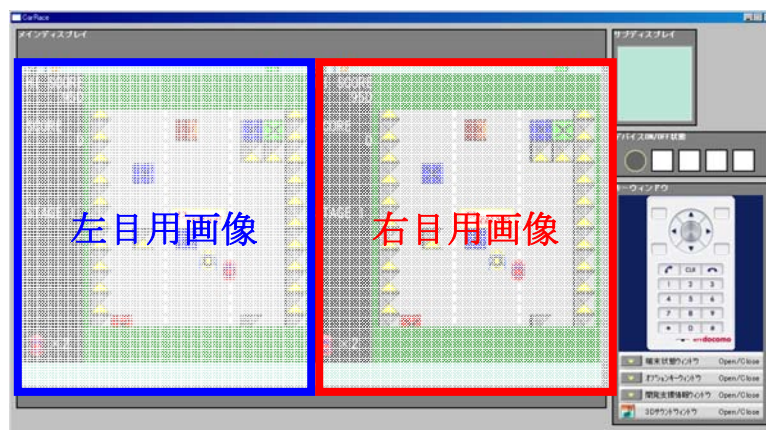


図 14-6 縦方向のステレオ表示（平行法）

これに対し、横方向を設定（StereoScope#setStereoscopicStyle()でSTEREOSCOPIC_STYLE_HORIZONTALを設定）した場合は、左 90 度回転した状態で横方向にステレオ表示されます。（図 14-7）

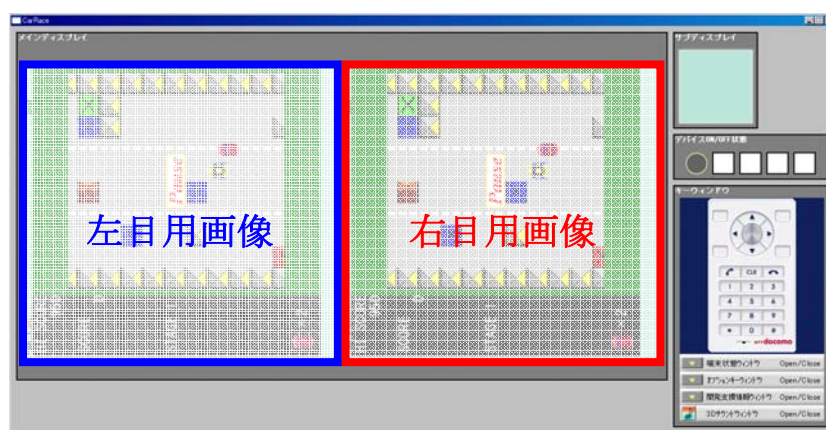


図 14-7 横方向のステレオ表示（平行法）

ただし、ディスプレイ状態が「右 90 度回転」に設定されている場合は、右 90 度回転した状態で横方向にステレオ表示されます。

第15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。

- ・メーラ起動ファイル
- ・バウンダリデータファイル
- ・メールアドレスファイル (※)
- ・ブラウザ起動HTMLファイル
- ・着信履歴ファイル
- ・メールファイル (※)
- ・電話帳ファイル (※)
- ・電話帳グループファイル (※)
- ・マルチメディア領域
- ・ブックマーク登録ファイル (※)
- ・コード認識結果ファイル
- ・LED表示点灯パターン登録ファイル
- ・ストレージデバイス ファイルパス
- ・内部ストレージデバイス ファイルパス
- ・トルカファイル (※)
- ・ソーステンプレートファイル
- ・位置情報ファイル (※)
- ・位置情報画像登録ファイル (※)
- ・測位データ設定ファイル (※)
- ・スクリーンショットファイル
- ・電子コンパスデータファイル (※)
- ・サウンド設定ファイル (※)
- ・仮想FeliCaアドホック通信設定ファイル
- ・Bluetooth設定ファイル (※)
- ・BML連携起動パラメータファイル (※)
- ・DTV連携起動パラメータファイル (※)
- ・視聴/録画予約情報ファイル (※)
- ・「アプリケーション名.properties」ファイル (※)
- ・仮想FeliCaデータファイル (※)
- ・スクラッチパッドデータファイル (※)
- ・ハイパーマルチデータファイル (※)
- ・加速度センサデータファイル (※)
- ・動作認識データファイル (※)
- ・音声認識データファイル (※)
- ・対応フォントサイズ設定ファイル (※)
- ・署名者情報ファイル
- ・CSV入出力ファイル
- ・歩数計ファイル (※)
- ・歩数計設定ファイル
- ・ケータイ払いデータファイル
- ・ポイント払いデータファイル
- ・個別課金コンテンツ権利データファイル
- ・都市情報設定ファイル
- ・コピー文字列起動パラメータファイル (※)
- ・身体活動量計データファイル (※)
- ・オートGPS制御情報データファイル
- ・ジャイロセンサデータファイル (※)
- ・身体情報データファイル (※)
- ・脈拍計データファイル (※)
- ・赤外線受信データファイル
- ・音楽プレイヤー関連ファイル (※)
- ・RAW画像取得機能関連ファイル
- ・開発支援情報表示設定ファイル (※)

※ このファイルは、iα ppliTool（画面）から設定できます。

15.1. メーラ起動ファイル

ファイルの場所:

C:¥iDKStar2.0¥lib¥mail¥*.mail

メーラ起動ファイルは、「4.16. メーラ起動」で使します。

サンプルで、C:¥iDKStar2.0¥lib¥mail¥0.mail 1.mail が用意されています。

メーラ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 (StarApplication)	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

15.2. バウンダリデータファイル

ファイルの場所:

C:¥iDKStar2.0¥lib¥mail¥boundary.dat

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。

15.3. メールアドレスファイル

ファイルの場所:

C:¥iDKStar2.0¥lib¥mail¥myAddress.dat

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。

15.4. ブラウザ起動 HTML ファイル

ファイルの場所:

Web サーバの適切な場所

ブラウザ起動HTMLファイルは、「4.17. ブラウザ起動」、「4.34 MyFACE起動」で使します。

サンプルで、C:¥iDKStar2.0¥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。

15.5. 着信履歴ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥phonehistory¥phonehistory.lst

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 ID
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 ID 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。

15.6. メールファイル

ファイルの場所:

C:\¥iDKStar2.0¥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

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 (StarApplication)	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（「7.4. 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 ID になります。

15.7. 電話帳ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥phonebook¥[エントリ ID]_[名前].tel

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 ID
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 ID
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

15.8. 電話帳グループファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥group¥[グループ ID]_[グループ名].grp

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 ID
2	GROUP1	グループ名

15.9. マルチメディア領域

ディレクトリの場所:

C:\iDKStar2.0\lib\multimedia

i アプリから ImageStore.addEntry() などで、マルチメディア領域へ登録すると、上記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。

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 appli Development Kit for Star-2.0」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。

15.10. ブックマーク登録ファイル

ファイルの場所:

C:\iDKStar2.0\lib\bookmark\[エントリ ID].bmk

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	http://xxx.xxx.xxx/xxx	URL
2	bookmark	ブックマーク名

15.11. コード認識結果ファイル

ファイルの場所:

C:\iDKStar2.0\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。

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。

「i appli Development Kit for Star-2.0」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（「15.9. 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) JAN8 コードの場合

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

2) JAN13 コードの場合

XXXXXXXXXXXXXX	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 コードは以下のとおりです。 (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- AV:[12345678901234567890123456];	TV 発信用電話番号 (複数指定可) 《TV 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

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

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

※ 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】

「LED表示点灯パターン登録ファイル」に格納される内容は、以下に示すとおりです。

333

15.13. ストレージデバイス ファイルパス

ファイルの場所:

C:¥iDKStar2.0¥lib¥storagedevice¥xxx

ストレージデバイスでアクセスするファイルパスです。

File オブジェクトは、このフォルダ上のファイルにアクセスします。

ファイルは、**ext0/SD_BIND/SVCXXXXYYYYZZZZ** 配下に作成されます。

XXXX: 共有識別とアクセス識別の値

YYYY: アプリケーション共有の場合: ADF のファイル名

CP 共有の場合: トラステッド APID (指定が無ければ 11 桁の 0)

ZZZZ: アクセス識別

UIM: レジストリの StorageAccessUIM の値

機器: レジストリの StorageAccessModel の値

シリーズ: レジストリの StorageAccessSeries の値

シリーズと機器が指定された場合は機器の値になります。

15.14. 内部ストレージデバイス ファイルパス

ファイルの場所:

C:¥iDKStar2.0¥lib¥internalstoragedevice¥

内部ストレージデバイスでアクセスするファイルパスです。

InternalStorageFile オブジェクトは、このフォルダ上のファイルにアクセスします。

インストール後、初回の InternalStorage.getMyFolder() メソッドの実行時に、このフォルダ配下に **int0** フォルダが作成され、このフォルダが内部ストレージデバイスのルートフォルダ

(/) となります。InternalStorage.getMyFolder() メソッドにより自アプリ固有フォルダの新規作成を行うと、このフォルダ配下にアプリケーション名と同名のフォルダが追加されます。

15.15. トルカファイル

ファイルの場所:

C:¥iDKStar2.0¥lib¥toruca¥toruca[エントリ ID].dat

トルカのデータが格納されたファイルで、トルカ起動でも使われます。

トルカファイルは、次の様なファイルで作成します。

torucaxxxxxxxxxx.dat (xxxxxxxx: エントリ ID)
--

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=xxxxxx	データ 1 (Base64 形式)
6	Data2=xxxxxx	データ 2 (Base64 形式)
7	Data3=xxxxxx	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.jam" type ="application/x- jam"></OBJECT> <CENTER> SampleToruCa CarRace 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 7.8 トルカデータ編 集の機能を用いて編集を行う ように下さい。(自動的 に置換・保存されます。)

あらか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。

15.16. ソーステンプレートファイル

ファイルの場所:

C:¥iDKStar2.0¥lib¥template¥xxx.txt

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、「4.3.1. ソースファイルの作成」を参照してください。

15.17. 位置情報ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥location¥location_information¥[エントリ ID].loc

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

各カラム間カンマ区切りで指定します。

15.18. 位置情報画像登録ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥location¥location_information¥[エントリ ID].img

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

各カラム間カンマ区切りで指定します。

15.19. 測位データ設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥location¥survey_data¥[通番].srv

C:\¥iDKStar2.0¥lib¥location¥survey_data¥survey_order (順序)

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	1～3の数値	0～2147483647	0(GPS測位方式)固定

			は不明を示す)			
--	--	--	---------	--	--	--

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 1 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。

15.20. スクリーンショットファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥screenshot¥scr_[通番].bmp

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。

「8.8. 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。

15.21. 電子コンパスデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥location¥compass.dat

電子コンパスの情報が格納されたファイルです。

「電子コンパスデータファイル」に格納される内容は、以下に示すとおりです。

磁北から直面 Y 軸方向の方位角	磁北から直面 X 軸方向の方位角
実数(整数部 3 桁、小数部 6 桁)	実数(整数部 3 桁、小数部 6 桁)
[0 度, 360 度)	[0 度, 360 度)

各カラム間はカンマ区切りで指定します。

Compass#getAzimuth() の実行時（ワンショット計測時）は、実行するたびに 1 行目から順に
 データを 1 件分取得して Degree オブジェクトに格納し返却します。ファイルの最後まで取得
 したら、1 行目に戻って取得します。

Compass#getData() 実行時（周期計測時）は、Compass#start() によって計測を開始した時点か
 ら Compass#getData() を実行するまでの時間経過と、start() で指定された計測間隔、および
 「アプリケーション名.properties」ファイルに設定された電子コンパスデータの最大保存件
 数・計測間隔解像度を考慮した件数分、1 行目から順に取得して配列変数に格納し返却します。
 次回実行時は前回取得した次の行から取得します。ファイルの最後まで取得したら、1 行目
 に戻って取得します。

PhoneSystem.setAttribute() によって設定される PhoneSystem.DEV_EYE_DIRECTION 属性の設
 定値によって、以下の値を返却します。

DEV_EYE_DIRECTION 属性の値	返却される値
ATTR_EYE_DIRECTION_Y_PLUS	「Y 軸方向の方位角」
ATTR_EYE_DIRECTION_X_PLUS	「X 軸方向の方位角」

ATTR_EYE_DIRECTION_Y_MINUS	「Y 軸方向の方位角」の 180 度反転
ATTR_EYE_DIRECTION_X_MINUS	「X 軸方向の方位角」の 180 度反転

(「180 度反転」…180 未満の場合は+180、180 以上の場合は-180)

15.22. サウンド設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥SoundConf.properties

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	MFi のチャンネル 1 の代替音
:	:	:
16	CH_16=-1	MFi のチャンネル 16 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

15.23. 仮想 FeliCa アドホック通信設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥FelicaAdhoc.properties

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 回で受け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。

15.24. Bluetooth 設定ファイル

Bluetooth 通信機能に関する設定およびデータが格納されるファイルです。

15.24.1. Bluetooth 通信設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥bluetooth¥bluetooth.ini

Bluetooth 通信の設定情報、および登録機器リストが格納されたファイルです。

Bluetooth 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	[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)/血圧計(7)/体温計(8)
3	deviceAddr=112233445566	自デバイスの Bluetooth アドレス
4	deviceName=StarBluetooth	自デバイスの名称
5	specData=123456	自デバイスの Spec Data
6	localName=LocalName	自デバイスの Local Name
7	uuid=1234	自デバイスの UUID
8		
9	[ScanningDevice]	接続先の設定
10	maxConnection=3	同時に接続可能な Bluetooth デバイス数の上限値
11	host=127.0.0.1	Bluetooth 通信の相手先ホスト名 複数設定する場合は「,」で区切る(最大 3 件まで)
12	basePort=1674	Bluetooth 通信の基準ポート番号
13		
14	[Device]	外部デバイス(接続相手)の情報
15	count=3	外部デバイスの件数
16		
17	[Device_1]	外部デバイス 1 のデータ 外部デバイスを示すキー「Device_?」の?の部分は、1 から COUNT までであり、途中省略できません。
18	deviceAddr=112233445566	外部デバイス 1 の Bluetooth アドレス ここで指定した Bluetooth アドレスに対応する Bluetooth デバイス情報設定ファイルが必ず存在しなければいけません。また、登録される Bluetooth アドレスは一意である必要があります。
19	spp=false	外部デバイス 1 の SPP プロファイル対応状況 (Inquiry 結果) true : SPP で接続可能 false : SPP で接続不可能 spp、hdp、registered の少なくともいずれか 1 項目が true である必要があります。
20	hdp=true	外部デバイス 1 の HDP プロファイル対応状況 (MDEP 設定状況) true : HDP で接続可能 false : HDP で接続不可能 true の場合、対応する MDEP 設定ファイルが存在している必要があります。 spp、hdp、registered の少なくともいずれか 1 項目が true である必要があります。
21	registered=false	外部デバイス 1 の機器登録状況 true : 登録済み false : 未登録 spp、hdp、registered の少なくともいずれか 1 項目が true である必要があります。
22		
23	[Device_2]	外部デバイス 2 の情報

:	:	:
---	---	---

エミュレータの疑似 Bluetooth 通信機能で使用する TCP ポート番号は以下のとおりです。i アプリで TCP/UDP 通信機能を使用しており、Bluetooth 通信機能で使用するポート番号と競合する場合は、basePort の値を変更することで競合を回避できます。

basePort から始まる番号（デフォルト値=1674～1676）：Page Scan 機能および SPP 接続時の connect / accept で使用

basePort+maxConnection から始まる番号（デフォルト値=1677～1679）：Inquiry Scan 機能で使用

basePort+maxConnection×2 から始まる番号（デフォルト値=1680～1682）：シンプルペアリング機能で使用

15.24.2. Bluetooth デバイス情報設定ファイル

ファイルの場所：

C:\¥iDKStar2.0¥lib¥bluetooth¥[Bluetooth アドレス].ini

外部デバイスに関する詳細情報が格納されたファイルです。

Bluetooth通信設定ファイルに「外部デバイス」として登録されているBluetoothアドレスに対応します。Bluetooth通信設定ファイルに「外部デバイス」が登録されている場合は、このファイルが必要となります。

Bluetooth デバイス情報設定ファイルのフォーマットは以下のとおりです。

1	[Device]	デバイスの情報
2	deviceType=516	デバイスの端末属性 PC(260)/携帯電話(516)/血圧計(7)/体温計(8)
3	deviceName=StarBluetooth	デバイスの名称
4	specData=123456	デバイスの Spec Data
5	localName=LocalName	デバイスの Local Name
6	uuid=1234	デバイスの UUID

これらの値は次のメソッドで取得できます。

RemoteDevice#getAddress()：Bluetooth アドレス

RemoteDevice#getDeviceName()：デバイスの名称

RemoteDevice#getDeviceClass()：デバイスの端末属性

DeviceDiscoveryResult#getAddress()：Bluetooth アドレス

DeviceDiscoveryResult#getServiceClassUUID16()：デバイスの UUID ※

DeviceDiscoveryResult#getMajorServiceClasses()：デバイスの端末属性（Major Service Classes）

DeviceDiscoveryResult#getMajorDeviceClass()：デバイスの端末属性（Major Device Classes）

DeviceDiscoveryResult#getMinorDeviceClass()：デバイスの端末属性（Minor Device Classes）

DeviceDiscoveryResult#getShortenedLocalName()：デバイスの Local Name

DeviceDiscoveryResult#getManufacturerSpecificData()：デバイスの Spec Data

DeviceDiscoveryResult#getDeviceName()：デバイスの名称

※ getServiceClassUUID16()メソッドで取得できる int[]配列の第 1 要素の値として取得できます。

15.24.3. 起動可能外部機器設定ファイル

ファイルの場所：

C:\iDKStar2.0\lib\bluetooth\remotelaunch.ini

起動可能外部機器リストが格納されるファイルです。
起動可能外部機器設定ファイルのフォーマットは以下のとおりです。

1	[Device]	外部デバイス(接続相手)の情報
2	count=3	外部デバイスの件数
3		
4	[Device_1]	外部デバイス 1 のデータ 外部デバイスを示すキー「Device_?」の?の部分は、1 から COUNT までであり、途中省略できません。
5	deviceAddr=112233445566	外部デバイス 1 の Bluetooth アドレス ここで指定したBluetoothアドレスは、Bluetooth通信設定ファイルに「外部デバイス」として登録され、かつregisteredがtrueに設定されている必要があります。
6	projectName=CarRace	外部デバイス 1 を起動可能外部機器として登録している i アプリのプロジェクト名 エミュレータのプロジェクト格納フォルダ配下に配置されるプロジェクトフォルダのフォルダ名と一致する必要があります。
7		
8	[Device_2]	外部デバイス 2 の情報
:	:	

これらの値は、i アプリから StarApplicationManager#addRemoteLauncher() を実行することにより登録され、StarApplicationManager#removeRemoteLauncher()を実行することにより削除されます。また StarApplicationManager#getRemoteLauncherList()によって登録状況を確認できます。

なお、実機では一つの外部デバイスを複数のアプリが起動可能外部機器として登録することはできませんが、エミュレータではそのような制限はありません。

15.24.4. MDEP 設定ファイル

ファイルの場所:

C:\iDKStar2.0\lib\bluetooth\hdp\Bluetooth アドレス\mdep.dat

Bluetooth HDP デバイスの MDEP (MCAP Data Access Point) に関する設定が格納されるファイルです。

Bluetooth通信設定ファイルに「外部デバイス」として登録され、かつhdp=trueに設定されている場合は、このファイルが必要となります。

MDEP 設定ファイルのフォーマットは以下のとおりです。

ID	Type	Role
数値(0~255 の整数値)	血圧計(7) /体温計(8)	Source(0) /Sink(1)

各カラム間はカンマ区切りで指定します。

これらの値は次のメソッドで取得できます。なお MCAPDataEndPoint オブジェクトは HDPCConnection# getMDEPList()によって取得します。

MCAPDataEndPoint#getMDEPID() : ID
 MCAPDataEndPoint#getMDEPDataType() : Type
 MCAPDataEndPoint#getMDEPRole() : ID

15.24.5. MDEP 入力データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥bluetooth¥hdp¥[Bluetooth アドレス]¥[MDEP ID]¥input.dat

Bluetooth デバイスに対して HDP で接続し、HDPCConnection#openInputStream() メソッドによって InputStream を取得すると、このファイルに記録されているデータをバイナリデータとして取得できます。

初期状態では、0 バイトのファイルが作成されています。

15.24.6. MDEP 出力データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥bluetooth¥hdp¥[Bluetooth アドレス]¥[MDEP ID]¥output[YYYYMMDDhhmmss]_[通番].dat

Bluetooth デバイスに対して HDP で接続し、HDPCConnection#openOutputStream() メソッドによって取得した OutputStream へ出力されたデータは、このファイルへバイナリデータとして書き出されます。

15.25. BML 連携起動パラメータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥BML.properties

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 StarApplicationManager.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。

15.26. DTV 連携起動パラメータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥DTV.properties

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 StarApplicationManager.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

StarApplicationManager#getPushedEvent(STAR_CALLED_BY_DTV) 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、

StarApplicationManager#getPushedEvent(STAR_CALLED_BY_DTV)は NULL を返します。

15.27. 視聴/録画予約情報ファイル

ファイルの場所:

C:¥iDKStar2.0¥lib¥schedule¥[エントリ ID].dtv

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)

15.28. 「アプリケーション名.properties」ファイル

ファイルの場所:

C:\¥iDKStar2.0¥apps¥[アプリケーション名]¥[アプリケーション名].properties

携帯電話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	accelerationDevice = 2	加速度センサーの有効デバイス数
11	accelerationDataMax = 20	加速度センサデータの最大保存件数

12	bookmarkEntry = 10	ブックマークの最大保存件数
13	launchType = 0	i アプリ読み込み時の起動タイプ
14	groupEntry = 10	グループの最大保存件数
15	dtvScheduleMax = 50	視聴/録画予約の最大保存件数
16	remoteParameter =	リモート制御の受信パラメータ
17	remotePushedTime =	リモート制御の受信時間
18	sound3dResource = 4	3D サウンドの制御を行うためのリソース数
19	cameraDevice = 1	カメラデバイス数
20	cameraFocusMode = 0	カメラのフォーカス機構 (0:ソフトウェア 1:ハードウェア)
21	launchTime =	i アプリのスケジューラ起動時刻 n@YYYY/MM/DD/hh/mm/ss, XX, XX n:設定番号 1 ~ XX:Span Set ; で区切って複数指定
22	launchTimeMax = 4	スケジューラ起動時刻の最大数 (4 固定)
23	fontBaselinePos =	フォントベースライン設定
24	stubSourceName = XXX	i アプリライブラリのスタブファイル名
25	componentVendor = XXX	i アプリライブラリの会社名
26	componentTitle = XXX	i アプリライブラリの名前
27	componentName = XXX	i アプリライブラリの JAR ファイル名
28	componentAuthor = XXX	i アプリライブラリの作者
29	componentVersion = XXX	i アプリライブラリのバージョン
30	componentCreate = yes	i アプリライブラリの有効/無効 (true / false)
31	tcpudpwait = 0	TCP/UDP 通信で遅延させる時間(ミリ秒)
32	invitationParameter =	招集メッセージパラメータ
33	invitationResponse = true	招集メッセージ通信結果
34	gyroSensorMaxDataSize = 3000	ジャイロセンサデータの最大保存件数
35	gyroSensorIntervalResolution = 100	ジャイロセンサデータの計測間隔解像度(ミリ秒)
36	gyroSensorCalibrateTime = 10000	ジャイロセンサのキャリブレーションタイムアウト値(ミリ秒)
37	SchedulerParam =	スケジューラ連携起動パラメータ
38	DataBoxParam = XXX	データ BOX 画像連携起動時の起動元画像ファイルパス
39	compassMaxDataSize = 3000	電子コンパスデータの最大保存件数
40	compassIntervalResolution = 100	電子コンパスデータの計測間隔解像度(ミリ秒)
41	daysForSavingPedometerDataMax = 30	歩数計データの最大保存日数
42	maxByte = 128	音声合成機能で音声出力する文字列データに設定できる最大バイト数(Byte)
43	maxRuby = 16	音声合成機能でルビ情報として登録できる最大のデータ数
44	readSpeed = 200	音声合成機能の文字の読み上げ速度(ミリ秒)
45	contentsPackageParam =	パッケージ起動パラメータ

15.29. 仮想 FeliCa データファイル

ファイルの場所:

C:\¥iDKStar2.0¥apps¥[アプリケーション名]¥[アプリケーション名].fc

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
	SystemDefineBlock	1	先頭 2 バイトは、SystemCode
	AreaDefineBlock	1	先頭 4 バイトは、Area0000
フリーエリア 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	
フリー領域 フリーエリア 2	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	
フリーエリア 3	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 = 16 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（「9.3. 仮想FeliCaデータファイルビューワ」を参照）。

15.30. スクラッチパッドデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥apps¥[アプリケーション名]¥sp¥[アプリケーション名].sp

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、「9.1. スクラッチパッド管理ツール」を参照してください。

15.31. ハイパーマルチデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥HyperMulti.properties

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	セパレートモード: 1, 2, 3

15.32. 加速度センサデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥location¥acceleration¥[デバイス ID].dat

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。

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

各カラム間はカンマ区切りで指定します。

15.33. 動作認識データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥gesture¥roll¥[通番].ges

C:\¥iDKStar2.0¥lib¥gesture¥roll¥gesture_roll_order (順序)

C:\¥iDKStar2.0¥lib¥gesture¥shake¥[通番].ges

C:\¥iDKStar2.0¥lib¥gesture¥shake¥gesture_shake_order (順序)

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

15.34. 音声認識データファイル

音声認識機能で利用するデータが格納されるファイルです。

15.34.1. マスター設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥recog¥settinEntry.dat

選択された音声認識データのエントリ ID が格納されるファイルです。

マスター設定ファイルの記述方法は、以下に示すとおりです。

1	0001	エントリ ID
2		EOF

15.34.2. 音声取得情報設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥recog¥[通番]¥recogSetting.dat

特微量データを自動生成するための各種設定値と音声認識情報が格納されるファイルです。

音声取得情報設定ファイルの記述方法は、以下に示すとおりです。

1	sample data	設定タイトル
2	0	DDP 詳細設定
3	200	ネゴシエーションレスポンス
4	200	送信レスポンス
5	1	送信回数
6	200	結果取得レスポンス
7	200	キャンセルレスポンス
8	200	終了レスポンス
9	AP-FDFE/1.0	FDFE 名称 ※
10	DCM-DSR	対応コーデック (固定値) ※
11	30000	最大発話時間 ※
12	140	フレームサイズ ※
13	240	1 フレームの時間(msec) ※ 特微量データを出力する間隔を指定
14	100	レディ時間 ※
15	42	パディング値 (42='*') ※ ASCII コード 10 進で指定
16	100	レベル ※
17	100	SN 比 ※
18	1	音声有無 ※
19		特微量定義ファイル名：空行固定 ※
20	recogResult.dat	認識結果情報ファイル名 ※
21		EOF

※ 設定ファイルを直接変更する必要があります。

15.34.3. 認識結果情報設定ファイル

ファイルの場所：

C:¥iDKStar2.0¥lib¥recog¥[通番]¥recogResult.dat

擬似 DDP 通信にて GET-RESULT コマンドが成功した場合に取得する情報が格納されるファイルです。

認識結果情報設定ファイルの記述方法は、以下に示すとおりです。

1	Y	認識結果
2	0	認識結果ステータス
3	N-BEST	結果タイプ (固定値)
4	2	認識結果
:	/駅は何処ですか?/, 90, /駅/は/何処/ですか?/ /駅はそこです/, 70, /駅/は/そこ/です/	ResultType 依存情報
n		EOF

15.35. 対応フォントサイズ設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥FontSize.ini

対応フォントサイズが格納されるファイルです。

「対応フォントサイズ設定ファイル」に格納される内容は、以下に示すとおりです。

1	[FONT_SIZE]	対応フォントサイズの情報
2	MAX_SIZE=60	対応フォントサイズの上限值 60～128 の範囲で指定します。

15.36. 署名者情報ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥Signature.properties

署名機能で利用する署名者情報が格納されるファイルです。

「署名者情報ファイル」に格納される内容は、以下に示すとおりです。

1	Common_Name = DoCoMo	一般名
2	Organization = NTT DoCoMo, Inc.	組織名
3	Organization_Unit = DoCoMo Secure Network Primary	部門名
4	SerialNumber = 75BCD15	シリアルナンバー (16進数)

これらの値は次のメソッドで取得できます。

PKCS7SignerInfo#getIssuer(COMMON_NAME) : 一般名

PKCS7SignerInfo#getIssuer(ORGANIZATION) : 組織名

PKCS7SignerInfo#getIssuer(ORGANIZATION_UNIT) : 部門名

PKCS7SignerInfo#getSerialNumber() : シリアルナンバー

15.37. CSV 入出力ファイル

ファイルの場所:

任意

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

15.38. 歩数計ファイル

歩数計機能で利用する歩数計データが格納されるファイルです。

15.38.1. 歩数計サマリーファイル

ファイルの場所：

C:\¥iDKStar2.0¥lib¥pedom¥summary.dat

全ての歩数計データのサマリー情報を日単位で保持するためのファイルです。

歩数計サマリーファイルの記述方法は、以下に示すとおりです。

日付	計測周期 (秒)	一日の歩 数	一日の歩 行距離	一日の有 酸素運動 の歩数	一日の有 酸素運動 の時間	一日の燃 焼脂肪量	一日の消 費カロリ ー
yyyy/mm/ dd	数値	数値	数値	数値	数値	数値	数値

各カラム間カンマ区切りで指定します。

15.38.2. 歩数計データファイル

ファイルの場所：

C:\¥iDKStar2.0¥lib¥pedom¥[日付].dat

一日分の歩数計データを計測周期単位で保持するためのファイルです。

歩数計データファイルの記述方法は、以下に示すとおりです。

歩数	歩行距離	有酸素運動 歩数	有酸素運動 時間	燃焼脂肪量	消費カロリ ー
数値	数値	数値	数値	数値	数値

各カラム間カンマ区切りで指定します。

1 行 1 周期とし、全体の行数は 86400 ÷ 計測周期 となります。

これらの値は次のメソッドで取得できます。

PedometerData#getCalendar () ： 対象日

PedometerData#getNumberOfSteps() ： 歩数

PedometerData#getDistance() ： 歩行距離

PedometerData#getNumberOfAerobicsSteps() ： 有酸素運動の歩数

PedometerData#getAerobicsTime() : 有酸素運動の時間
 PedometerData#getAmountOfCombustionFat() : 燃焼脂肪量
 PedometerData#getConsumptionCalorie() : 消費カロリー
 PedometerData#getInterval() : 計測周期

15.39. 歩数計設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥pedom¥pedometer.properties

歩数計機能の設定情報が格納されるファイルです。

「歩数計設定ファイル」に格納される内容は、以下に示すとおりです。

1	limit_connection = 1	ミニアプリからの同時接続数上限
---	----------------------	-----------------

15.40. ケータイ払いデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥contents¥paymentResponse.dat

ケータイ払い決済機能で使用する決済データが格納されるファイルです。

「ケータイ払いデータファイル」に格納される内容は、以下に示すとおりです。

1	[PHONE]	ケータイ払いの情報
2	ID=0123456789AB	決済番号 12桁の半角英数字を指定します。
3	DATE=20110101000000	決済日時 (YYYYMMDDHHMMSS 形式)

ケータイ払い決済機能を使用する i アプリにおいて、ケータイ払い決済対象のコンテンツの購入・決済申し込みを行うメソッド Contents.sendPaymentRequest()が成功した場合、戻り値として、上記に指定した内容が String[]形式で取得できます。

15.41. ポイント払いデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥contents¥pointExchangeResponse.dat

ポイント払い決済機能で使用する決済データが格納されるファイルです。

「ポイント払いデータファイル」に格納される内容は、以下に示すとおりです。

1	[POINT]	ポイント払いの情報
2	ID=0123456789AB	決済番号 12桁の半角英数字を指定します。
3	DATE=20110101000000	決済日時 (YYYYMMDDHHMMSS 形式)

ポイント払い決済機能を使用する i アプリにおいて、ポイント払い決済対象のコンテンツの購入・決済申し込みを行うメソッド Contents.sendPointExchangeRequest()が成功した場合、戻り値として、上記に指定した内容が String[]形式で取得できます。

15.42. 個別課金コンテンツ権利データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥contents¥purchaseResponse.dat

個別課金機能で使用する権利データが格納されるファイルです。

「個別課金コンテンツ権利データファイル」に格納される内容については、特に規定はありません。

個別課金機能を使用する i アプリにおいて、個別課金対象のコンテンツの購入・課金申し込みを行うメソッド `Contents.sendPurchaseRequest()` が成功した場合、戻り値として、「個別課金コンテンツ権利データファイル」の内容が `byte[]` 形式で取得できます。

15.43. 都市情報設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥cityInfoStore.properties

位置情報取得 i アプリが使用する、都市情報取得機能で取得できる情報が格納されるファイルです。

「都市情報設定ファイル」に格納される内容は、以下に示すとおりです。

1	cityinfo1=1000	都市情報エントリ ID
---	----------------	-------------

位置情報 i アプリが、ローミングアウト中に都市情報を取得するために `CityInfoStore.selectEntryId()` を呼び出したときの戻り値として、「都市情報設定ファイル」に格納される都市情報エントリ ID が取得可能です。

15.44. コピー文字列起動パラメータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥copystring.dat

コピー文字列起動時に、i アプリに渡される起動パラメータが格納されるファイルです。

「コピー文字列起動パラメータファイル」に格納される内容は、任意のテキストデータです。

ここに設定したテキストデータは、コピー文字列起動時に表示されるパラメータ入力画面に初期値として表示され、「起動」ボタンを押すことでパラメータ入力画面に設定された内容に更新されます。

15.45. 身体活動量計データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥physicalactivity.dat

身体活動量計機能で利用する身体活動量計データが格納されるファイルです。

「身体活動量計データファイル」に格納される内容は、以下に示すとおりです。

1	[PHYSICALACTIVITYMONITOR]	身体活動量計共通項目
2	METS=1.5	現在の運動強度
3	COUNT=1	身体活動量情報の数
4	LIMIT=1	ミニアプリからの同時接続数上限
5	[PHYSICALACTIVITYDATA_1]	身体活動量情報の対象日 1 のデータ 対象日を示すキー 「PHYSICALACTIVITYDATA_?」の?の部分は、1 から COUNT までであり、途中省略できません。
6	EXERCISE_GROSS=0.0	対象日 1 の通常エクササイズ総量 EXERCISE_?のうち、通常エクササイズ値を示すエントリの値の合計と一致する必要があります。
7	EXERCISE_AEROBICS_GROSS=0.0	対象日 1 の有酸素運動エクササイズ総量 EXERCISE_?のうち、有酸素運動エクササイズ値を示すエントリの値の合計と一致する必要があります。
8	INTERVAL=3600	対象日 1 の周期 86400 の約数であり、かつ 60 の倍数である必要があります。
9	YEAR=2009	対象日 1 の年
10	MONTH=01	対象日 1 の月
11	DATE=01	対象日 1 の日
12	EXERCISESIZE=48	対象日 1 のエクササイズ値の総エントリ数 通常エクササイズ値のエントリ数と有酸素運動エクササイズ値のエントリ数の合計になります。 $86400 \div [\text{INTERVAL の値}] \times 2$ と一致する必要があります。
13	EXERCISE_1=0	対象日 1 の第 1 周期の通常エクササイズ値 エクササイズ値を指定するキー 「EXERCISE_?」の?の部分は、1 から EXERCISESIZE までであり、途中省略できません。 前半分(左記例の場合 EXERCISE_1~24)が通常エクササイズ値、後半分(左記例の場合 EXERCISE_25~48)が有酸素運動エクササイズ値となります。
14	EXERCISE_2=0	対象日 1 の第 2 周期の通常エクササイズ値
:	:	
36	EXERCISE_24=0	対象日 1 の第 24 周期の通常エクササイズ値
37	EXERCISE_25=0	対象日 1 の第 1 周期の有酸素運動エクササイズ値
38	EXERCISE_26=0	対象日 1 の第 2 周期の有酸素運動エクササイズ値
:	:	
60	EXERCISE_48=0	対象日 1 の第 24 周期の有酸素運動エクササイズ値
61	[PHYSICALACTIVITYDATA_2]	身体活動量情報の対象日 2 のデータ

:	:	:
---	---	---

これらの値は次のメソッド（オプション）で取得できます。

PhysicalActivityMonitor#getMETs() : 運動強度
 PhysicalActivityMonitor#getCount() : 身体活動量情報の数
 PhysicalActivityData#getCalendar() : 日付
 PhysicalActivityData#getInterval() : 周期
 PhysicalActivityData#getExercise(...) : エクササイズ値

15.46. オート GPS 制御情報データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥atfdata.dat

AutoPositioning#setParameter()メソッドによって i アプリから登録されたオート GPS 制御情報が格納されるファイルです。

「オート GPS 制御情報データファイル」に格納される内容は、以下に示すとおりです。

1	[ATFDATAPROPERTY]	オート GPS 機能共通項目
2	COUNT=3	オート GPS 制御情報の数 オート GPS 制御情報のデータ件数 （[ATFDATA_?]セクションの数）と一致する必要があります。
3	[ATFDATA_1]	1 件目のオート GPS 制御情報のデータ 制御情報を示すキー「ATFDATA_?」の?の部分、1 から COUNT までであり、途中省略できません。
4	PROJECT_NAME =AppName	1 件目のオート GPS 制御情報に対応する自動測位利用アプリのプロジェクト名 エミュレータのプロジェクト格納フォルダ配下に配置されるプロジェクトフォルダのフォルダ名と一致する必要があります。
5	APP_NAME= AppName	1 件目のオート GPS 制御情報に対応する自動測位利用アプリ名
6	URL=http://localhost/atfserv er/	1 件目のオート GPS 制御情報に対応する位置情報通知先 URL
7	DESCRIPTION=	1 件目のオート GPS 制御情報に対応するフリー領域
8	STOP_TIME=1	1 件目のオート GPS 制御情報に対応する位置情報通知停止期間
9	STATUS=0	1 件目のオート GPS 制御情報の状態（エミュレータでは未使用）
10	[ATFDATA_2]	2 件目のオート GPS 制御情報のデータ
:	:	

PROJECT_NAME(プロジェクト名)は、i アプリから AutoPositioning#getParameter()メソッド、AutoPositioning#delete()メソッドが呼び出された際に、呼び出した i アプリに対応するオート GPS 制御情報を特定するために使用します（i アプリから PROJECT_NAME の設定値を参照することはできません）。

15.47. ジャイロセンサデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥gyrosensor.dat

ジャイロセンサ機能で利用するジャイロセンサデータが格納されるファイルです。

「ジャイロ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	[GYRODATAPROPERTY]	ジャイロセンサ共通項目
2	COUNT=2	ジャイロセンサ情報の数 ジャイロセンサデータのデータ件数 ([GYRODATA_?])セクションの数)と一致する必要があります。
3	[GYRODATARANGE]	角速度範囲設定
4	MINVALUE_X=-180	X軸角速度範囲の下限值(deg/s)
5	MAXVALUE_X=180	X軸角速度範囲の上限値(deg/s)
6	MINVALUE_Y=-180	Y軸角速度範囲の下限值(deg/s)
7	MAXVALUE_Y=180	Y軸角速度範囲の上限値(deg/s)
8	MINVALUE_Z=-180	Z軸角速度範囲の下限值(deg/s)
9	MAXVALUE_Z=180	Z軸角速度範囲の上限値(deg/s)
10	[GYRODATA_1]	1件目のジャイロセンサデータ ジャイロセンサデータを示すキー 「GYRODATA_?」の?の部分、1 から COUNT までであり、途中省略できません。
11	VALUE_X=90	1件目のジャイロセンサデータの X 軸角速度(deg/s) 設定値は $\text{MINVALUE_X} \leq \text{VALUE_X} \leq \text{MAXVALUE_X}$ である必要があります。
12	VALUE_Y=90	1件目のジャイロセンサデータの Y 軸角速度(deg/s) 設定値は $\text{MINVALUE_Y} \leq \text{VALUE_Y} \leq \text{MAXVALUE_Y}$ である必要があります。
13	VALUE_Z=90	1件目のジャイロセンサデータの Z 軸角速度(deg/s) 設定値は $\text{MINVALUE_Z} \leq \text{VALUE_Z} \leq \text{MAXVALUE_Z}$ である必要があります。
14	[GYRODATA_2]	2件目のジャイロセンサデータ
:	:	

これらの値は次のメソッド（オプション）で取得できます。

GyroSensor#getMinDataValue() : 角速度範囲の下限值

GyroSensor#getMaxDataValue() : 角速度範囲の上限値

GyroData#getAngularVelocityX() : X 軸角速度

GyroData#getAngularVelocityY() : Y 軸角速度

GyroData#getAngularVelocityZ() : Z 軸角速度

GyroSensor#getCurrentData() の実行時（ワンショット計測時）は、実行するたびに [GYRODATA_1] から順にデータを 1 件分取得して GyroData オブジェクトに格納し返却します。ファイルの最後まで取得したら、[GYRODATA_1] に戻って取得します。

GyroSensor#getData() 実行時（周期計測時）は、GyroSensor#start() によって計測を開始した時点からGyroSensor#getData() を実行するまでの時間経過と、start() で指定された計測間隔、および「アプリケーション名.properties」ファイルに設定されたジャイロセンサデータの最大保存件数・計測間隔解像度を考慮した件数分、[GYRODATA_1]から順に取得してGyroDataオブジェクトに格納し返却します。次回実行時は前回取得時の次の[GYRODATA_n]から順に取得します。ファイルの最後まで取得したら、[GYRODATA_1]に戻って取得します。

15.48. 身体情報データファイル

ファイルの場所：

C:\¥iDKStar2.0¥lib¥bodyprofiles.dat

身体情報アクセス機能で利用する身体情報データが格納されるファイルです。

「身体情報データファイル」に格納される内容は、以下に示すとおりです。

1	[BODYPROFILE]	身体情報
2	HEIGHT_VALUE=2100	身長 (mm)
3	WEIGHT_VALUE=60000	体重 (g)
4	[BODYPROFILESETTING]	身体情報設定項目
5	HEIGHT_RESOLUTION=5	身長の分解能(mm)。1以上の整数値を設定する必要があります。
6	WEIGHT_RESOLUTION=500	体重の分解能(g)。1以上の整数値を設定する必要があります。
7	HEIGHT_MIN=0	身長の最小値(mm)。身長の分解能(HEIGHT_RESOLUTION)の倍数である必要があります。
8	HEIGHT_MAX=2100	身長の最大値(mm)。身長の分解能(HEIGHT_RESOLUTION)の倍数である必要があります。
9	WEIGHT_MIN=0	体重の最小値(g)。体重の分解能(WEIGHT_RESOLUTION)の倍数である必要があります。
10	WEIGHT_MAX=200000	体重の最大値(g)。体重の分解能(WEIGHT_RESOLUTION)の倍数である必要があります。
:	:	

これらの値は次のメソッド（オプション）で取得できます。

BodyProfiles.getBodyHeight()：身長

BodyProfiles.getBodyWeight()：体重

BodyProfiles.getAttribute(DEV_HEIGHT_RESOLUTION)：身長の分解能

BodyProfiles.getAttribute(DEV_WEIGHT_RESOLUTION)：体重の分解能

BodyProfiles.getAttribute(DEV_HEIGHT_MIN)：身長の最小値

BodyProfiles.getAttribute(DEV_HEIGHT_MAX)：身長の最大値

BodyProfiles.getAttribute(DEV_WEIGHT_MIN)：体重の最小値

BodyProfiles.getAttribute(DEV_WEIGHT_MAX)：体重の最大値

15.49. 脈拍計データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥pulsemeter¥pulsemeter.dat

脈拍計機能で利用する脈拍計データが格納されるファイルです。

「脈拍計データファイル」に格納される内容は、以下に示すとおりです。

1	-1	脈拍数 1
2	100	脈拍数 2
:	:	:
N	120	脈拍数 n

これらの値は次のメソッド（オプション）で取得できます。

実行するたびに 1 行目から順次データを取得して返します。ファイルの最後まで取得したら、1 行目に戻って取得します。

Pulsemeter#getPulserate()

15.50. 赤外線受信データファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥irreceiver¥ir_receive[通番].dat

赤外線受信機能で利用する赤外線受信データが格納されるファイルです。

「赤外線受信データファイル」に格納される内容は、以下に示すとおりです。

1	E10011100101010001010100F	受信データ 1
2	E0100010001001111101000011010011110100110101001111F	受信データ 2
:	:	:
N	E10011100101010001010100F	受信データ n (最大 50 行)

これらの値は、オプション機能である IrReceiver#receive()メソッドを使用して赤外線リモコン信号を受信した上で、その IrReceiver オブジェクトから取り出した InputStream を使用して取得することができます。

IrReceiver#receive()を実行すると赤外線受信データファイルのひとつ（ir_receive[通番].dat のいずれか）を読み込みます。読み込んだ内容は IrReceiver#getInputStream()により取得される InputStream に対してファイルの先頭から 500 ミリ秒間隔で 1 行ずつ渡され、InputStream#read()により赤外線受信データファイルのデータを取得することができます。ファイルの EOF に到達したら InputStream へのデータの引渡しは完了します。

なお、IrReceiver#receive()が最初に実行されるときに読み込まれるファイルは ir_receive1.dat であり、以降このメソッドを呼び出す度に ir_receive2.dat・・・と通番の順に読み込まれます。最後の通番のファイルまで読み込んだら ir_receive1.dat に戻ります。

15.51. 音楽プレイヤー関連ファイル

音楽プレイヤー機能で利用するデータが格納されるファイルです。

15.51.1. 音楽コンテンツデータファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥avplayer¥contents¥[音楽コンテンツファイル名].properties

音楽コンテンツごとの情報を保持するためのファイルです。

音楽コンテンツデータファイルの記述方法は、以下に示すとおりです。

1	fileFormat = 0	ファイル形式・種別 フル楽曲(アラカルトコンテンツ) : 0 フル楽曲(サブスクリプションコンテンツ) : 1 WindowsMediaAudio : 2 MPEG Audio Layer-3 : 3 SD-Audio : 4
2	title = Title1	タイトル
3	artist = Artist1	アーティスト
4	album = Album1	アルバム
5	year = 2010	年
6	genre = genre1	ジャンル
7	comment = comment1	コメント
8	trackNumber = 1	トラック番号 (トラック番号) 1以上の整数値
9	totalTrackNumber = 10	トラック番号 (総トラック番号) 1以上の整数値
10	composer = composer1	作曲者
11	lyricist = lyricist1	作詞者
12	rightsHolder = rightsholder1	権利者
13	sellingAgency = sellingagency1	販売元
14	rightsInfo = rightsinfo1	権利情報
15	label = label1	レーベル
16	url = urlname1	URL 情報
17	alturl = alturlname1	URL 情報 (URL 代替文字列)
18	ringToneTypeFull = true	着信音設定可否 (まるごと着信音) true : 設定可 false : 設定不可
19	ringToneTypeRecommended = true	CP 指定位置着信音設定可否 (オススメ着 信音) true : 設定可 false : 設定不可
20	ringToneTypeFree = true	任意箇所選択着信音設定可否 (おこのみ着 信音) true : 設定可 false : 設定不可
21	restrictCount = false	再生制限 (回数) true : 制限あり false : 制限なし
22	restrictTerm = false	再生制限 (期間) true : 制限あり

		false : 制限なし
23	restrictTermStart = 2010/01/01 00:00:00	再生可能期間 (開始日時) (YYYY/MM/DD HH:MM:SS 表記)
24	restrictTermEnd = 2010/01/01 00:00:00	再生可能期間 (終了日時) (YYYY/MM/DD HH:MM:SS 表記)
25	restrictLimit = false	再生制限 (期限) true : 制限あり false : 制限なし
26	restrictLimitEnd = 2030/01/01 00:00:00	再生可能期限 (YYYY/MM/DD HH:MM:SS 表記)
27	uimRestrict = 0	UIM 制限 UIM による制限がない : 0 UIM が挿入されていないことによる制限がある : 1 異なる UIM が挿入されていることによる制限がある : 2
28	gracePeriod = 2040/01/01 00:00:00	コンテンツ更新猶予期限 (YYYY/MM/DD HH:MM:SS 表記)
29	isFileRestricted = false	ファイル制限 true : 制限あり false : 制限なし
30	jacketImageFileName = jacket.gif	ジャケット画像 ※拡張子は.gif、.jpg、.jpeg のいずれか
31	isSavedJacketImage = true	保存可能 CD ジャケット画像 true : 保存可 false : 保存不可
32	standbyImage1FileName = standby1.gif	待ち受け画像 1 ※拡張子は.gif、.jpg、.jpeg のいずれか
33	isSavedStandbyImage1 = true	保存可能待ち受け画像 1 true : 保存可 false : 保存不可
34	standbyImage2FileName = standby2.gif	待ち受け画像 2 ※拡張子は.gif、.jpg、.jpeg のいずれか
35	isSavedStandbyImage2 true	保存可能待ち受け画像 2 true : 保存可 false : 保存不可
36	codecName = codecname1	音・オーディオ (コーデック)
37	bitrate = 10001.0	音・オーディオ (ビットレート)
38	fileSize = 1024	ファイルサイズ
39	originated = OriginName1	取得元
40	storedDate = 2010/01/01 00:00:00	保存日時 (YYYY/MM/DD HH:MM:SS 表記)
41	importPermission = 0	外部メモリから本体への移動 移動不可 : 0 移動可能 : 1 同一機種間のみ移動可能 : 2
42	exportPermission = 0	本体から外部メモリへの移動 移動不可 : 0 移動可能 : 1

		同一機種間のみ移動可能：2
43	<code>seekable = true</code>	シーク可否 true：シーク可 false：シーク不可
44	<code>playTime = 60000</code>	整数値（ミリ秒） ※1以上

これらの値は次のメソッドで取得できます。

ManagedData#getTitleName()：タイトル (title)
 Content#getType()：ファイル形式・種別 (fileFormat)
 Content#getFileName()：音楽コンテンツファイル名（音楽コンテンツデータファイルのファイル名から拡張子「.properties」を除いた部分）
 Content#getTotalTime()：総再生時間 (playTime)
 Content#getOrigin()：取得元 (originated)
 Content#getSavedDate()：保存日時 (storedDate)
 Content#getFileSizeOnMemory()：ファイルサイズ (fileSize)
 Content#canSetRingtone()：着信音設定可否（まるごと着信音） (ringToneTypeFull)
 Content#isFileRestricted()：ファイル制限 (isFileRestricted)
 Content#getExportPermission()：本体から外部メモリへの移動 (exportPermission)
 Content#getImportPermission()：外部メモリから本体への移動 (importPermission)
 MusicContent#getArtist()：アーティスト (artist)
 MusicContent#getAlbum()：アルバム (album)
 MusicContent#getGenre()：ジャンル (genre)
 MusicContent#getTrackNumber()：トラック番号 (trackNumber)
 MusicContent#getTotalTracks()：総トラック番号 (totalTrackNumber)
 MusicContent#getYear()：年 (year)
 MusicContent#getLyricist()：作詞者 (lyricist)
 MusicContent#getComposer()：作曲者 (composer)
 MusicContent#getLabel()：レーベル (label)
 MusicContent#getComment()：コメント (comment)
 MusicContent#getUrl()：URL 情報 (url)
 MusicContent#getAltUrl()：URL 情報 (URL 代替文字列) (alturl)
 MusicContent#getRightsHolder()：権利者 (rightsHolder)
 MusicContent#getSellingAgency()：販売元 (sellingAgency)
 MusicContent#getRightsInfo()：権利情報 (rightsInfo)
 MusicContent#getJacketImage()：ジャケット画像 (jacketImageFileName)
 MusicContent#getStandbyImages()：待ち受け画像 (standbyImage1FileName、standbyImage2FileName)
 MusicContent#hasExtractableJacketImage()：保存可能 CD ジャケット画像 (isSavedJacketImage)
 MusicContent#hasExtractableStandbyImage()：保存可能待ち受け画像 (isSavedStandbyImage1、isSavedStandbyImage2)
 MusicContent#canSetRingtone()：着信音設定可否 (ringToneTypeFull、ringToneTypeRecommended、ringToneTypeFree)
 AudioCodecInfo#getCodecName()：音・オーディオ（コーデック） (codecName)
 AudioCodecInfo#getBitRate()：音・オーディオ（ビットレート） (bitrate)
 PlayableInfo#getRestrictType()：再生制限有無 (restrictCount、restrictTerm、restrictLimit)
 PlayableInfo#getPlayableSpanStart()：再生可能期間（開始日時） (restrictTermStart)

PlayableInfo#getPlayableSpanEnd() : 再生可能期間（開始日時）（restrictTermEnd）

PlayableInfo#getPlayablePeriod() : 再生可能期限（restrictLimitEnd）

PlayableInfo#getGracePeriod() : コンテンツ更新猶予期限（gracePeriod）

PlayableInfo#getUIMRestrict() : UIM 制限（uimRestrict）

iα ppliTool画面（「7.19 音楽コンテンツ編集」）からデータを生成する際、画面に設定項目のない項目についてはダミー値を設定しますが、音楽コンテンツデータファイルの内容を直接編集することによって任意の値を返却させることが可能です。

15.51.2. プレイリストファイル

ファイルの場所：

C:¥iDKStar2.0¥lib¥avplayer¥playlist¥[プレイリスト名].dat

プレイリストごとの情報を保持するためのファイルです。

プレイリストファイルの記述方法は、以下に示すとおりです。

1	Music01.wma.properties	1 曲目の音楽コンテンツの音楽コンテンツデータファイル名
2	Music02.mp3.properties	2 曲目の音楽コンテンツの音楽コンテンツデータファイル名
:	:	:
N	MusicNN.wma.properties	n 曲目の音楽コンテンツの音楽コンテンツデータファイル名

行の記述順序は再生順序と一致します。

15.51.3. レジューム情報ファイル

ファイルの場所：

C:¥iDKStar2.0¥lib¥avplayer¥ResumptionInfo.properties

音楽制御機能を終了した際の状態を保持するためのファイルです。

レジューム情報ファイルの記述方法は、以下に示すとおりです。

1	contentsIndex = 1	カレント音楽コンテンツのコンテンツインデックス
2	playPosition = 60	カレント音楽コンテンツの再生位置
3	volume = 100	音量（0～100）
4	playMode = 1	再生モード 通常再生：0 シングルリピート：1 オールリピート：2 シャッフル：3 シャッフルリピート：4 1 曲終了：5
5	playlist = Playlist1.dat	カレントプレイリストのプレイリストファイル名 ※既存プレイリスト再生中の終了の場合
6	filterCount = 3	フィルタ条件数 ※フィルタ設定再生中の終了の場合 ※playlist と filterCount のいずれか一方必須

7	<code>filter.1.resultType = 1</code>	※filterCount 指定時 1 番目のフィルタ条件の結果タイプ 0 : アーティスト 1 : アルバム 2 : ジャンル 3 : コンテンツ
8	<code>filter.1.searchType = 2</code>	※filterCount 指定時 1 番目のフィルタ条件の検索タイプ 0 : アーティスト 1 : アルバム 2 : ジャンル
9	<code>filter.1.searchString = 検索文字列</code>	※filterCount 指定時 1 番目のフィルタ条件の検索文字列
:	:	:
*	<code>filter.[n].resultType = 1</code>	※filterCount 指定時 n 番目のフィルタ条件の結果タイプ
*	<code>filter.[n].searchType = 2</code>	※filterCount 指定時 n 番目のフィルタ条件の検索タイプ
*	<code>filter.[n].searchString = 検索文字列</code>	※filterCount 指定時 n 番目のフィルタ条件の検索文字列

レジューム情報ファイルは音楽プレイヤーAPI によって音楽再生が終了した契機で作成され、次回音楽再生ソフトウェア開始時の初期状態となります。

15.51.4. 画像ファイル格納領域

ディレクトリの場所:

`C:¥iDKStar2.0¥lib¥avplayer¥image`

音楽コンテンツ情報から参照されるジャケット画像、待ち受け画像ファイルを格納する領域です。

ファイル名の指定はありませんが、拡張子は“.gif”、“.jpg”、“.jpeg”のいずれかでなければなりません。

15.52. RAW 画像取得機能関連ファイル

RAW 画像取得機能にて利用する設定情報が格納されるファイルです。

15.52.1. RAW 画像取得設定ファイル

ファイルの場所:

`C:¥iDKStar2.0¥lib¥rawimage¥rawImageCapture.dat`

エミュレータとしてサポートする RAW 画像のサイズと画角に関する仕様を設定するためのファイルです。

RAW 画像取得設定ファイルの記述方法は、以下に示すとおりです。

幅	高さ	水平画角	垂直画角
数値(整数)	数値(整数)	数値(実数)	数値(実数)

各カラム間はカンマ区切りで指定します。

これらの値は次のメソッドで取得できます。

RawImageCapture#getAvailableImageSizes() : 幅・高さ

RawImageCapture#getRawImageAngleSize() : 幅・高さに対する水平画角・垂直画角

15.52.2. JPEG 画像データファイル

ディレクトリの場所:

C:\¥iDKStar2.0¥lib¥rawimage

RAW 画像取得機能 API によって取得できるデータの元画像となる JPEG 画像データを格納する領域です。

ファイル名は任意ですが拡張子は“.jpg”または“.jpeg”のいずれかとしてください。

RAW 画像取得機能 API を使用する場合は、必ず事前に上記領域に JPEG 画像データファイルを格納してください。RawImageCapture#getRawImage() を呼び出すと、エミュレータは上記領域にある JPEG 画像データファイルを読み込み、所定の RAW データ形式に変換して返却します。JPEG 画像データファイルが複数存在する場合は、getRawImage() が呼び出される都度、ファイル名の順に JPEG 画像データファイルを読み込みます。最後のファイルを読み込むと、その後は最初のファイルに戻ります。

上記領域に格納するファイルの JPEG 画像のサイズは、RAW 画像取得設定ファイルに設定されている画像サイズのリストに含まれ、かつ i アプリから RawImageCapture#setImageSize() で設定した画像サイズと一致している必要があります。画像サイズの異なる JPEG 画像データを読み込むと、RawImageCapture#getRawImage() は DeviceException 例外を発生します。

15.52.3. カメラファインダ設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥rawimage¥CameraFinder.properties

カメラファインダ機能の動作設定を保持するファイルです。

カメラファインダ設定ファイルの記述方法は、以下に示すとおりです。

1	transparenceR = 33	透過設定色 (R)
2	transparenceG = 0	透過設定色 (G)
3	transparenceB = 0	透過設定色 (B)
4	drawingR = 186	カメラファインダ描画領域色 (R)
5	drawingG = 231	カメラファインダ描画領域色 (G)
6	drawingB = 217	カメラファインダ描画領域色 (B)

カメラファインダ機能 API によってビューファインダ画像透過を開始すると、エミュレータではファインダ表示部分（矩形領域）の透過設定色描画部位を描画領域色に塗り替えます（カメラ画像の透過描画は行いません）。

15.53. 開発支援情報表示設定ファイル

ファイルの場所:

C:\¥iDKStar2.0¥lib¥DevSupportInfo.ini

開発支援上表表示画面の設定が格納されるファイルです。

「開発支援情報表示設定ファイル」に格納される内容は、以下に示すとおりです。

1 : [CpuThresholdSetting]	稼働率閾値設定
---------------------------	---------

2	CpuThreshold=80	スタックトレースを行う稼働率の閾値(%)
3	StackTraceEnabled=true	稼働率が閾値を超えたときのスタックトレース出力可否
4	[GraphSetting]	グラフ表示設定
5	GraphUpdateInterval=500	グラフ表示更新間隔(ミリ秒)
6	GraphAssistLineStep=25	グラフ補助線の間隔(%)
7	CpuThresholdLineColor=16776960	稼働率グラフの閾値線の表示色 ※32bit カラーコードを整数値で指定
8	GraphLineColor_High=16711680	グラフの表示色 (高域値) ※32bit カラーコードを整数値で指定
9	GraphLineColor_Mid=16776960	グラフの表示色 (中域値) ※32bit カラーコードを整数値で指定
10	GraphLineColor_Low=65280	グラフの表示色 (低域値) ※32bit カラーコードを整数値で指定
11	CpuGraphColorThreshold_High=80	稼働率グラフ表示色の高域値(HIGH)の閾値(%)
12	HeapGraphColorThreshold_High=80	ヒープ使用量グラフ表示色の高域値(HIGH)の閾値(%)
13	SendGraphColorThreshold_High=80	パケット送信量グラフ表示色の高域値(HIGH)の閾値(%)
14	ReceiveGraphColorThreshold_High=80	パケット受信量グラフ表示色の高域値(HIGH)の閾値(%)
15	PowerGraphColorThreshold_High=80	電力消費量グラフ表示色の高域値(HIGH)の閾値(%)
16	CpuGraphColorThreshold_Mid=50	稼働率グラフ表示色の中域値(MID)の閾値(%)
17	HeapGraphColorThreshold_Mid=50	ヒープ使用量グラフ表示色の中域値(MID)の閾値(%)
18	SendGraphColorThreshold_Mid=50	パケット送信量グラフ表示色の中域値(MID)の閾値(%)
19	ReceiveGraphColorThreshold_Mid=50	パケット受送信量グラフ表示色の中域値(MID)の閾値(%)
20	PowerGraphColorThreshold_Mid=50	電力消費量グラフ表示色の中域値(MID)の閾値(%)

15.54. ファイルの記述具体例

「http://xxx.yyy.zzz」の「CarRace.jam」をメーラ、ブラウザ、コードリーダー等から起動する場合:

- メーラ起動
 - 起動したい i アプリ (CarRace) の ADF の「LaunchByMail」に「any」を設定します。
 - メーラ起動ファイルに「C:¥iDKStar2.0¥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。
 - メーラ起動で、[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:¥iDKStar2.0¥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:/iDKStar2.0/lib/sample.html」と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 15.11. 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:¥iDKStar2.0¥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:¥iDKStar2.0¥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。

第16章 証明書の管理

「iα ppliTool」上で HTTPS 通信を行う際に使用する証明書は、OS 組み込みのものを使用します。よって、「iα ppliTool」では管理しません。

そこで、証明書を管理するためには OS にあらかじめインストールされている証明書管理アプリケーションを使用します。

1. 「コントロールパネル」→「インターネットオプション」をダブルクリックします（図 16-1）。

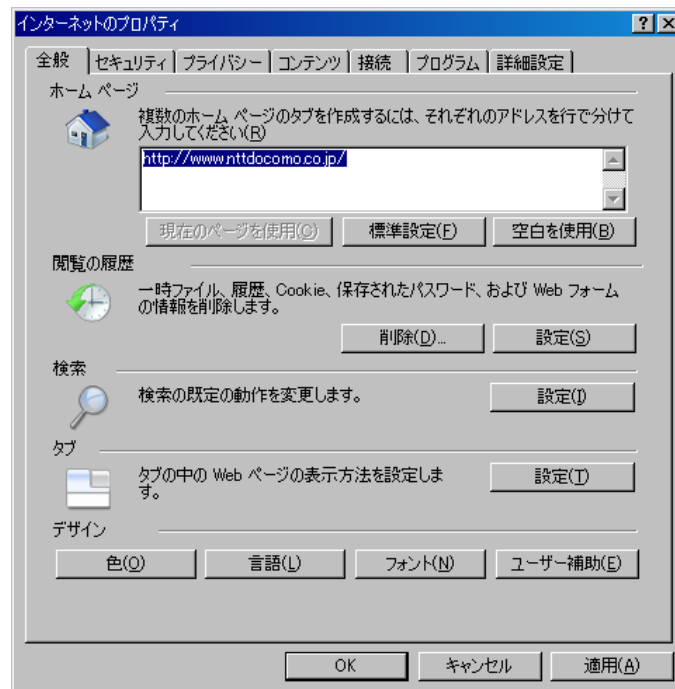


図 16-1 インターネットのプロパティ

2. 図 16-1 の「コンテンツ」タブを選択します（図 16-2）。

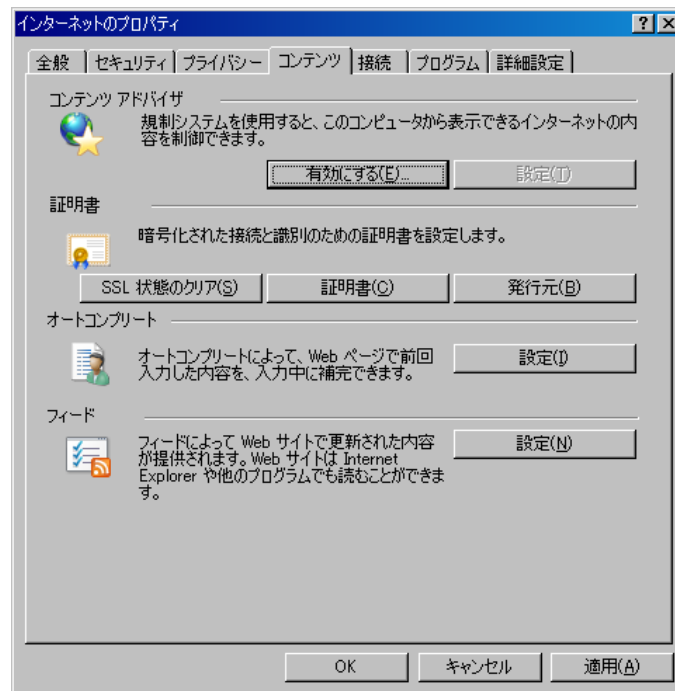


図 16-2 インターネットのプロパティ (コンテンツ)

- 図 16-2 の「証明書 (C) ...」ボタンを押します (図 16-3)。

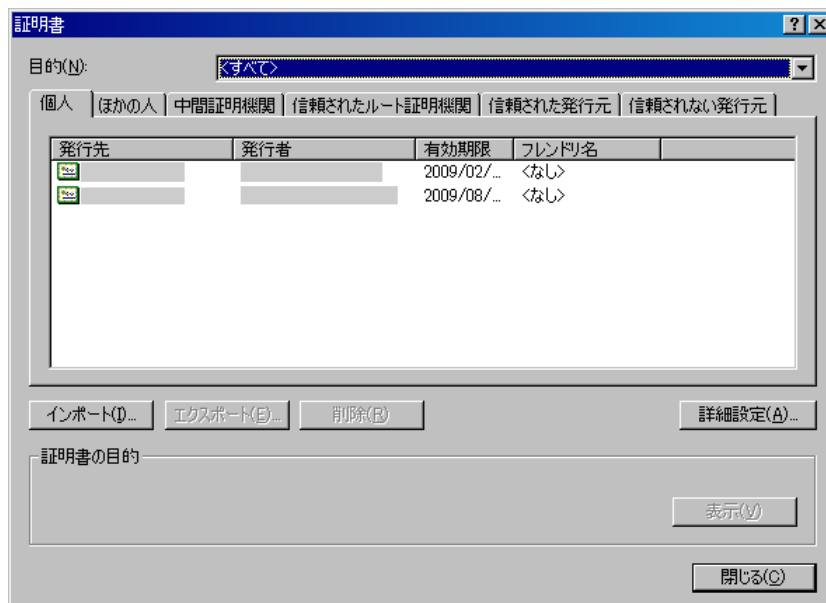


図 16-3 証明書画面

- 図 16-3 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17章 OBEX の利用

「iα ppli Development Kit for Star-2.0」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:\¥iDKStar2.0¥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。

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18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)、PNG、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19章 Eclipse の利用

この章では、Eclipse と「Eclipse プラグイン for iα ppli」について説明します。

19.1. 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、Eclipse 3.6 で使われる事を推奨します。
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 向けプラグインと共存させたい場合は、別の Eclipse 環境を用意してください。

「iα ppli Development Kit for Star-2.0」の Eclipse プラグイン for iα ppli で作成された i アプリ（JAR ファイル）は、Star-1.3 プロファイル搭載以降の携帯電話でのみ実行可能です。それ以前の携帯電話では i アプリを実行できません。

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「iα ppliTool」を使ってビルドしなおしてください。

Eclipse で i アプリにデバッガ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「デバッガによって使用される」チェックを全て外してください。
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。

19.2. 「Eclipse プラグイン for iα ppli」 のインストール

本節では、「Eclipse プラグイン for iα ppli」をインストールする手順を説明します。

本版の「Eclipse プラグイン for iα ppli」では、DoJa 向け i アプリを利用する事はできません。

DoJa 向けの i アプリを開発する場合は、別途、DoJa 向けの「Eclipse プラグイン for iα ppli」を用いてください。

19.2.1. 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 Development Kit (JDK) 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。

- JDK は、次のWeb サイトからダウンロードできます。
<http://www.oracle.com/technetwork/java/javase/overview/index.html>
➤ JDK のインストール後は、環境変数の PATH に JDK 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/>

19.2.2. 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、「Eclipse 3.6 プラグイン」をチェックします (図 19-1)。

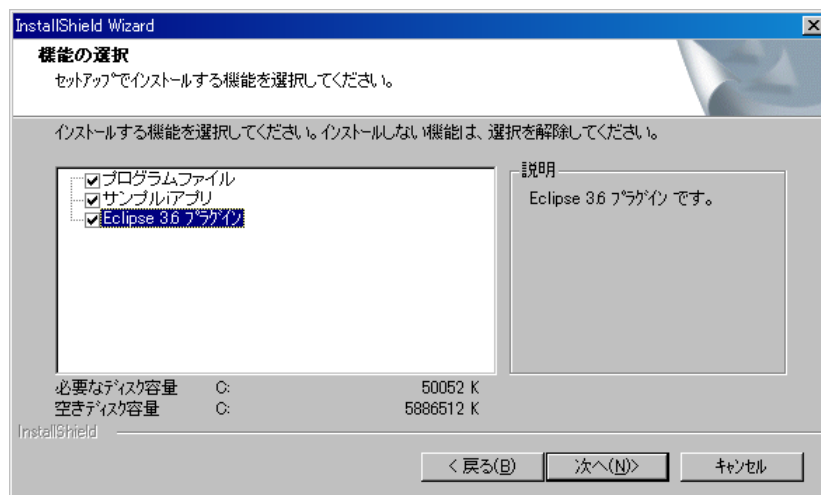


図 19-1 Eclipse プラグインのインストール

「Eclipse 3.6 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 (図 2-8)。ここで指定した Eclipse に対し、「Eclipse プラグイン for iα ppli」がインストールされます。

19.3. 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。

19.3.1. プロジェクトメニューのカスタマイズ

Eclipse で Star 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→「Star-2.0 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。

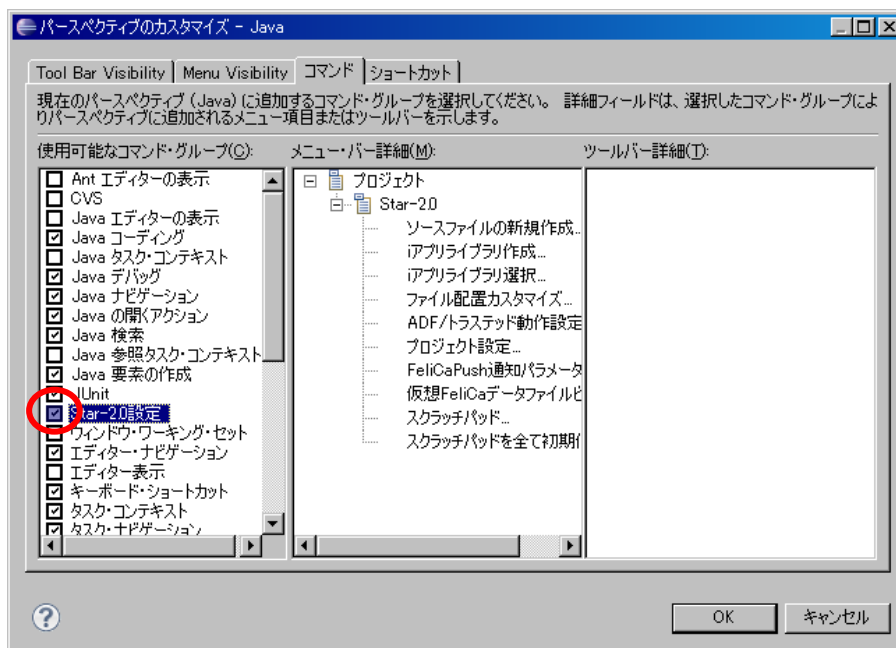


図 19-2 プロジェクトメニューのカスタマイズ

19.3.2. 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：

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「Star-2.0 プロジェクト」を選択し、「次へ」を押します。

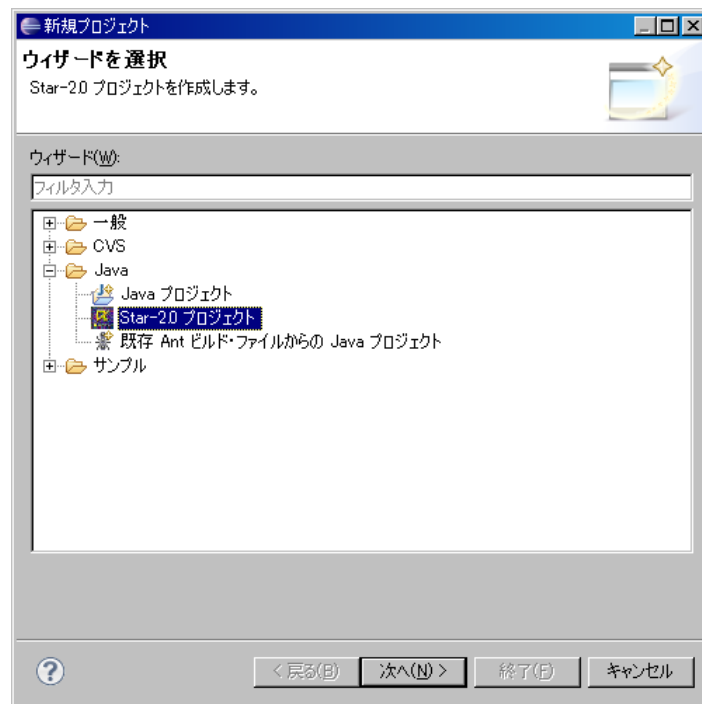


図 19-3 新規プロジェクトウィンドウ

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。

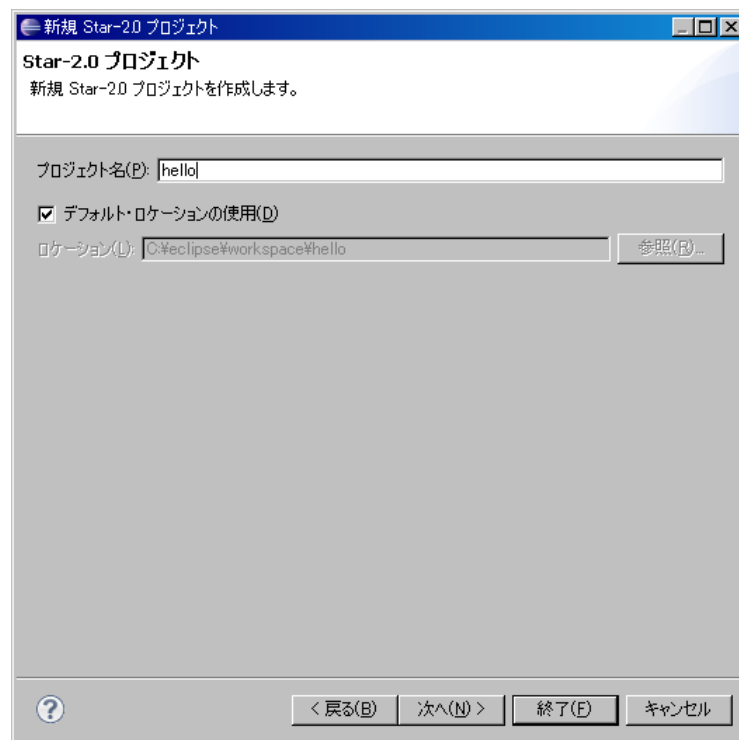


図 19-4 プロジェクト名設定ウィンドウ

- 「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「iα ppli Development Kit for Star-2.0」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. 次に作成するアプリケーションの種別を選択して「次へ」を押します。

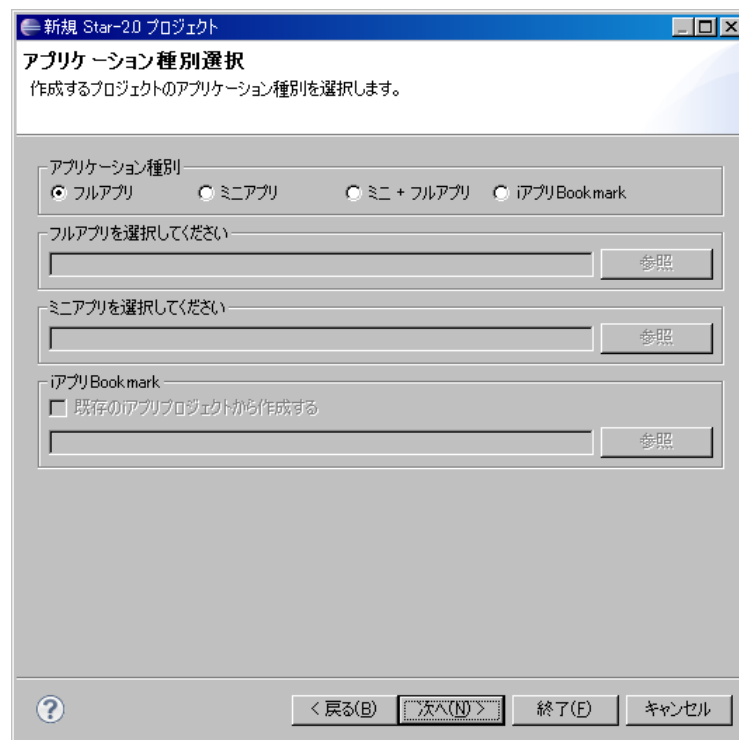


図 19-5 アプリケーション種別の選択

- 既存の i アプリプロジェクトから i アプリ Bookmark プロジェクトを作成する場合は、「既存の i アプリプロジェクトから作成する」にチェックした後、「参照」から対象の i アプリプロジェクトを選択してください。
5. 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。)
- 「ビルド・パス・エントリが欠落しています」と表示されることがありますが、この後の手順で自動的に作成されますのでそのまま進んでください。

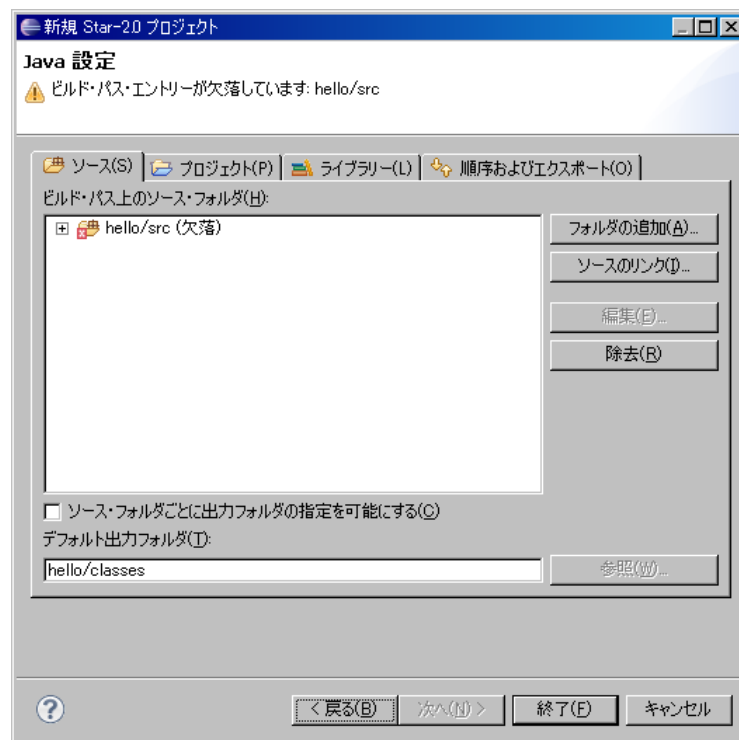


図 19-6 Java 設定ウィンドウ

6. パッケージ・エクスプローラービューにプロジェクト名 (hello) が追加されます。

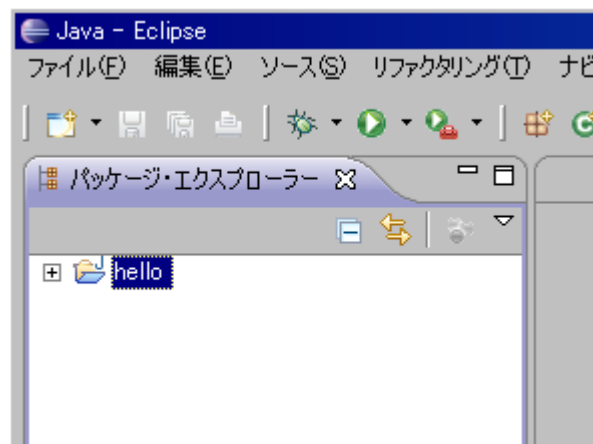


図 19-7 パッケージ・エクスプローラービュー

19.3.3. 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：

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。

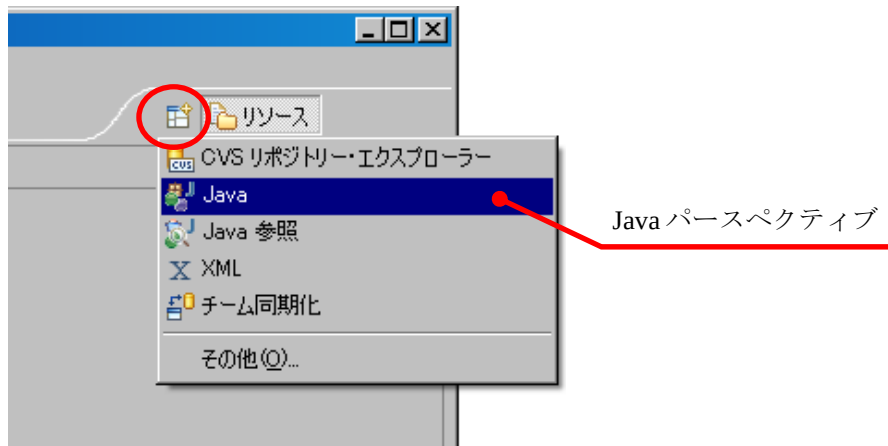


図 19-8 Java パースペクティブの選択

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 19-9)。

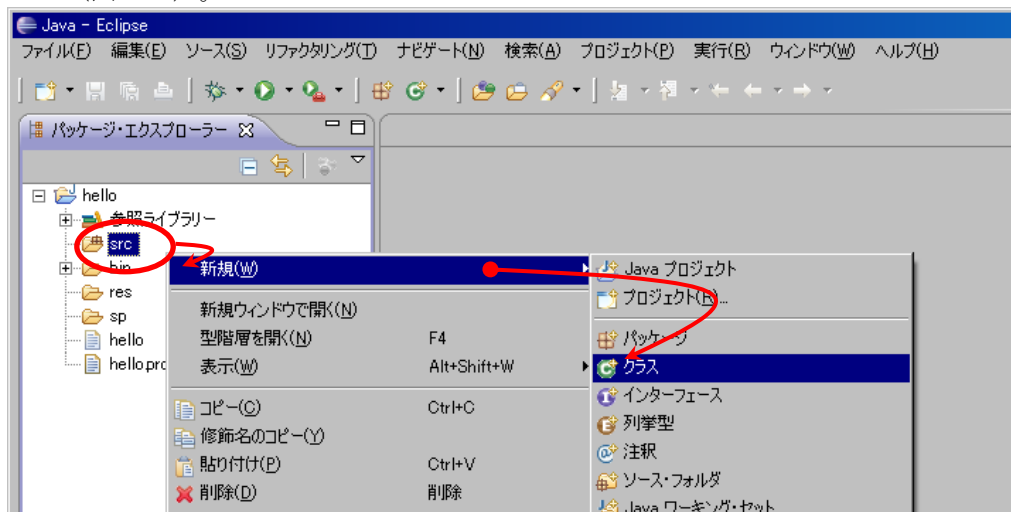


図 19-9 新規作成クラスの選択

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、スーパークラスに com.docomostar.StarApplication と入力します。

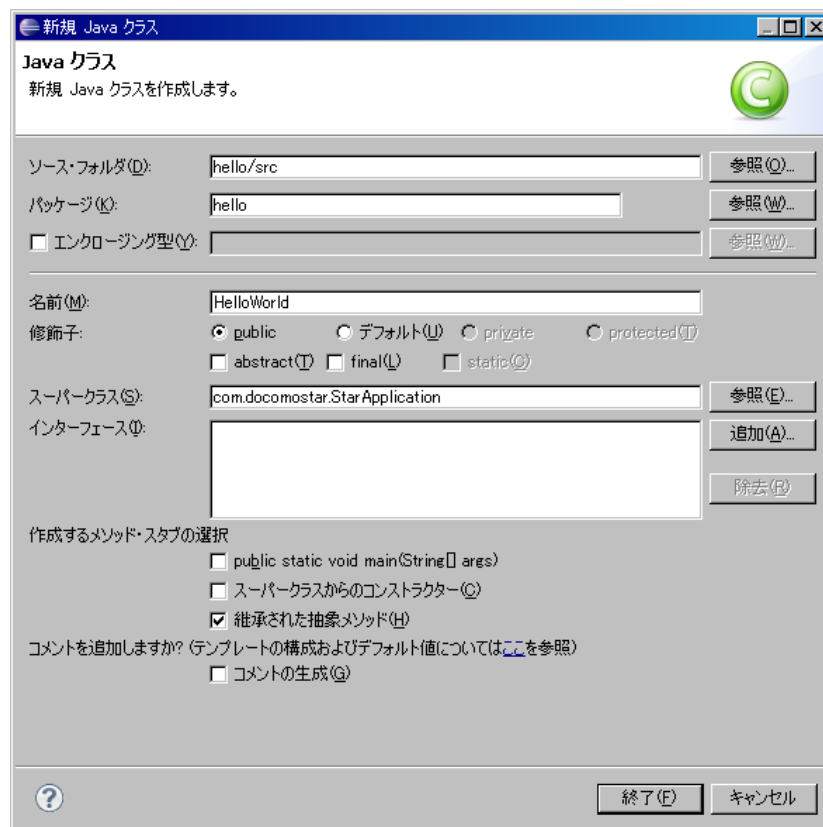


図 19-10 新規クラス設定ウィンドウ

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
public void started(int arg0) { の後に
System.out.println("Hello World");
を追加します (図 19-11)。

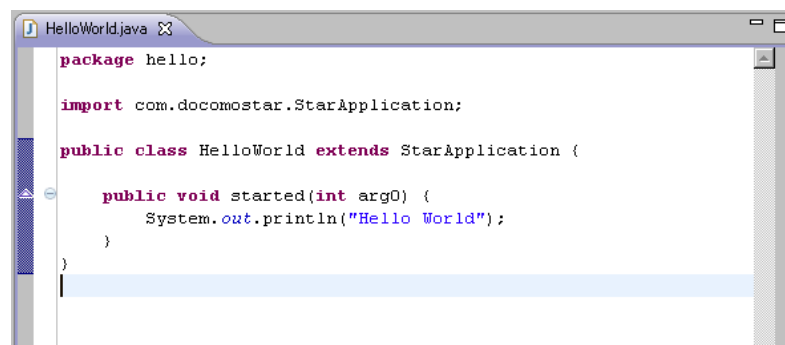


図 19-11 ソースファイルの編集

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。

19.3.4. 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、次の手順からも行えます。
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」→「Star-2.0」→「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。

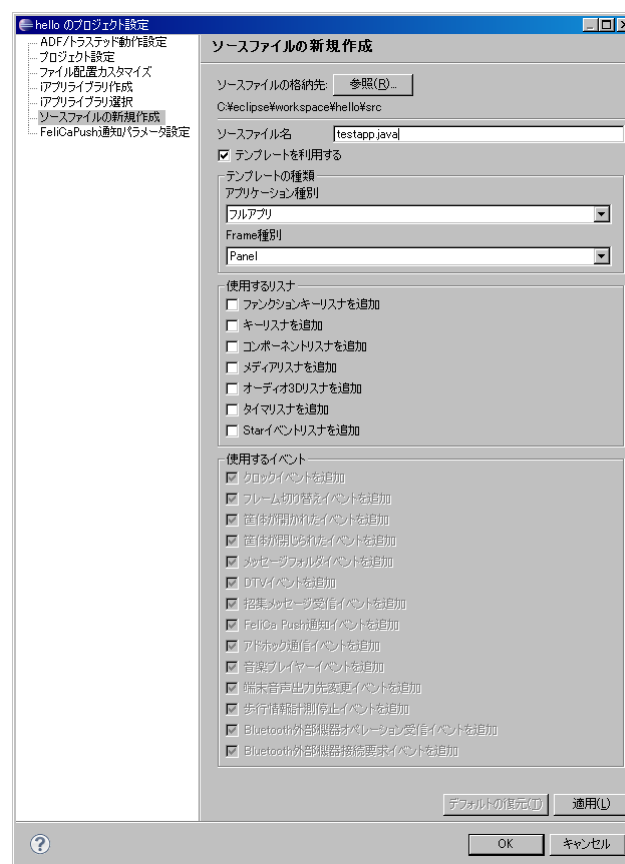


図 19-12 ソースファイルの新規作成画面

各ソースファイルの新規作成の設定項目については、「4.3.1 ソースファイルの作成」を参照してください。

19.3.5. 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：

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」→「Star-2.0」→「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。

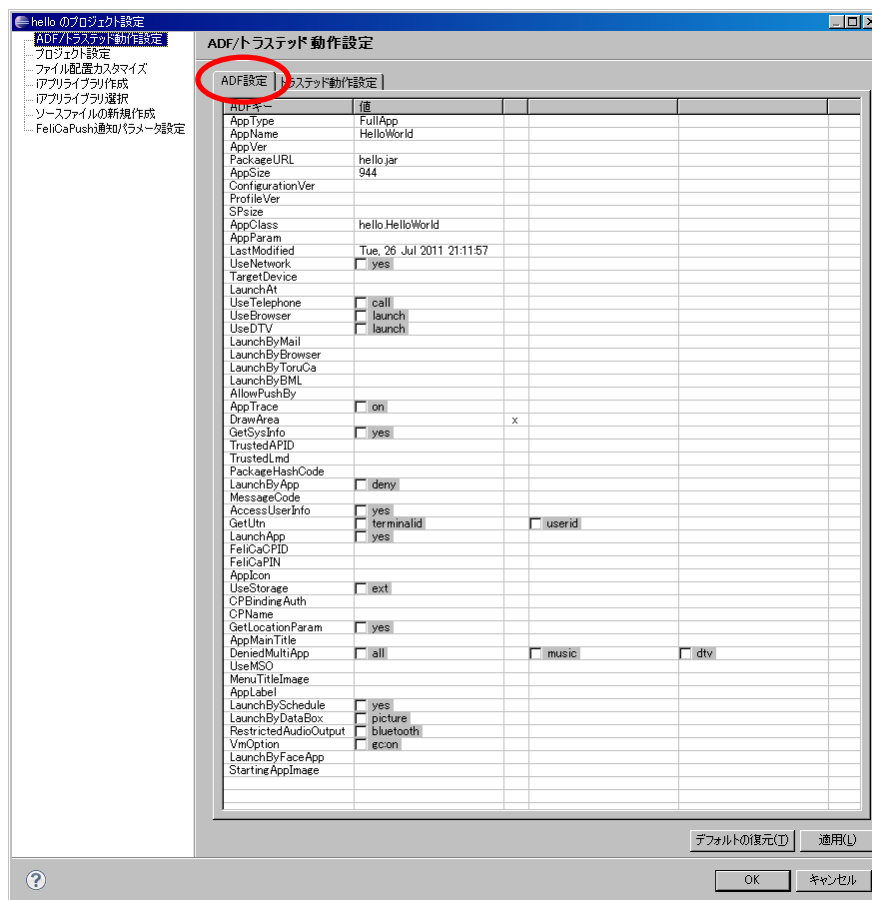


図 19-13 ADF 設定画面

各ADFの設定項目については、「4.8. アプリケーションディスクリプタファイル (ADF) の編集」（i アプリBookmarkプロジェクトについては「5.3 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。

19.3.5.1. 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 19-14)。



図 19-14 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。

19.3.5.2. 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 19-15）。

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。

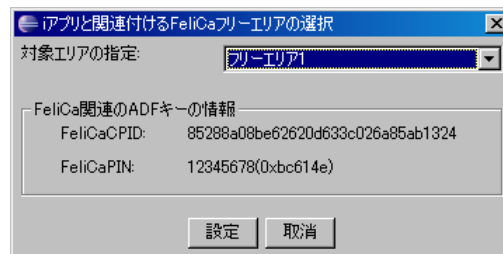


図 19-15 フリーエリアの選択

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。

19.3.6. 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：

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」→「Star-2.0」→「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、「トラステッド動作設定」タブをクリックし、必要に応じて各キーの属性値を変更してください。

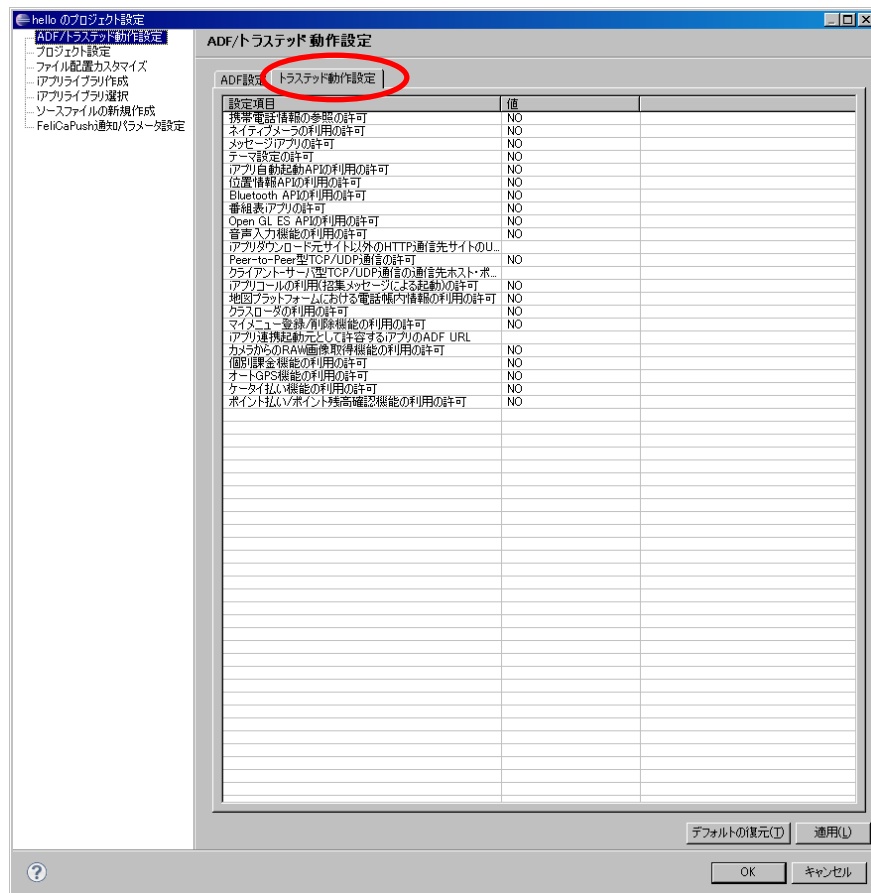


図 19-16 トラステッド動作設定画面

各トラステッド動作設定の設定項目については、「4.9. トラステッド動作の設定」を参照してください。

19.3.7. アプリケーションの構築と実行

「Eclipse プラグイン for iα ppli」では、i アプリを実行する際に、自動的に JAR ファイルの作成を行います。

※ i アプリ Bookmark プロジェクトは、「実行」操作を行えません。i アプリ Bookmark プロジェクトの場合は、プロジェクト更新操作（ADF の編集、res 配下へのファイルの追加）の都度、自動的に JAR ファイルを生成します。

アプリケーションの構築および実行をするには：

1. 「実行」→「実行構成...」（もしくは「実行」）を押します。
2. 実行構成ウィンドウが表示されるので、実行構成の「Star-2.0 アプリケーション」をマウスでダブルクリック、もしくは、新規作成ボタンを押します。
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。
Star タブでは、「プロジェクト」、「端末」、「スケール」、「デバッグトレース設定」、「デバッグモード起動」が設定可能です（図 19-17）。
引数タブに切り替えると、「起動種別」、「エントリ ID」、「サービス ID」、「画像ファイル名」、「BML 起動パラメータ」、「アプリケーション起動パラメータ」が設定可能です（図 19-18）。
4. 「プロジェクト」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. 「端末」でエミュレーション端末の種類を選択します。
6. 「スケール」で表示サイズを指定します。
7. 「デバッグトレース設定」でデバッグトレース出力の設定を指定します。
8. 「起動種別」で起動方法を選択します。
9. 「エントリ ID」で、位置情報起動、位置情報画像起動、位置情報電話帳起動時のエントリ ID を指定します。
10. 「サービス ID」で、DTV 起動時のサービス ID を指定します。
11. 「画像ファイル名」で、データ BOX 画像起動時の画像ファイル名を選択します。
12. 「BML 起動パラメータ」で、BML 起動時のパラメータを指定します。
13. 「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、メーラ起動・ブラウザ起動・コードリーダー起動・トルカ起動・招集メッセージ起動・コピー文字列起動・スケジューラ起動・MyFACE 起動時のパラメータも指定可能です。
なお、コピー文字列起動時は改行の入力も可能です。
14. 「実行」を押します。

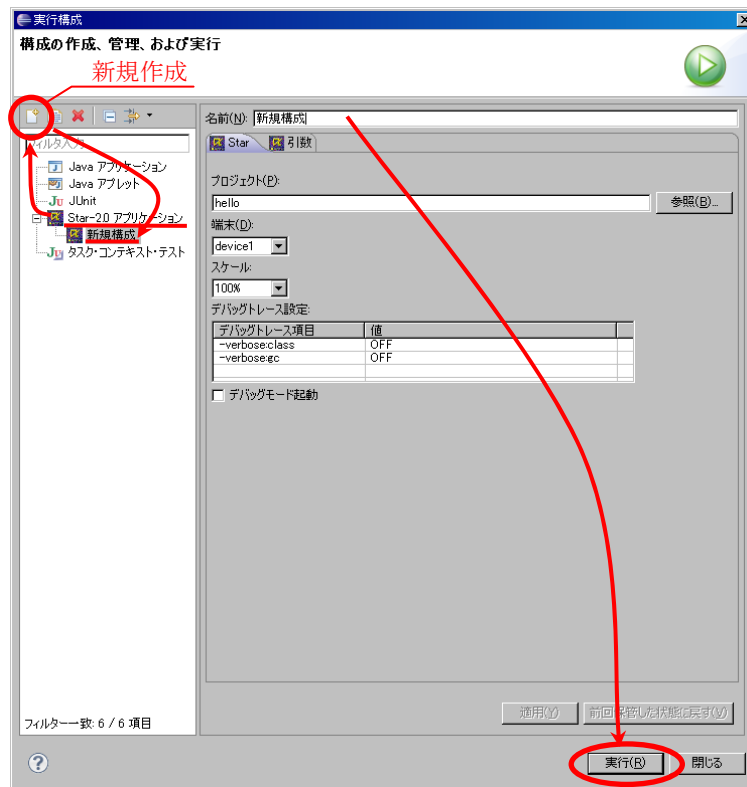


図 19-17 起動構成ウィンドウ (Star タブ)

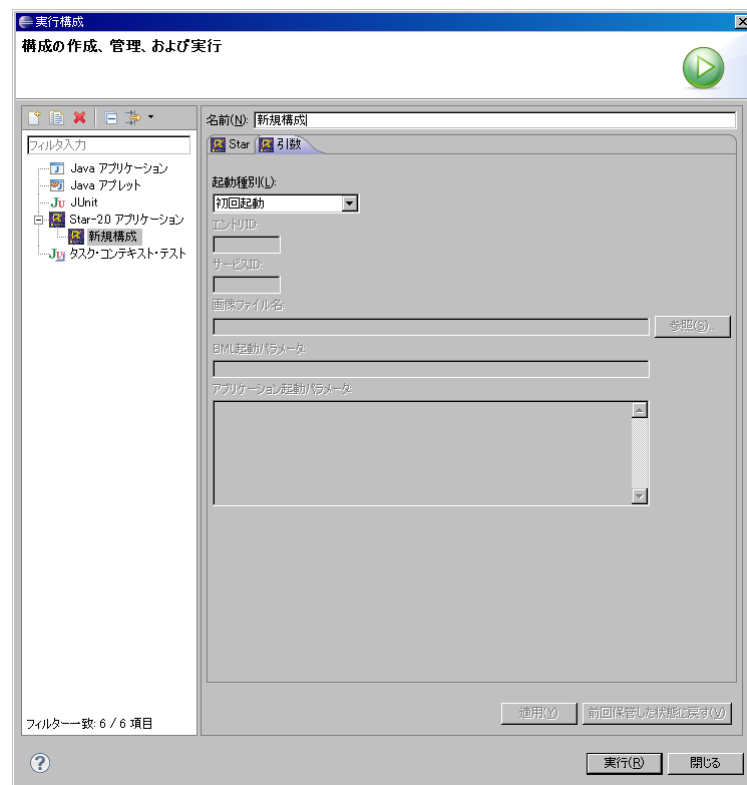


図 19-18 起動構成ウィンドウ (引数タブ)

15. 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。

※ 「iα ppli Development Kit for Star-2.0」の Eclipse プラグイン for iα ppli で作成した JAR ファイルは、Star-1.3 プロファイル搭載以降の携帯電話でのみ実行可能です。それ以前の携帯電話では実行できませんのでご注意ください。

16. 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 19-19）。

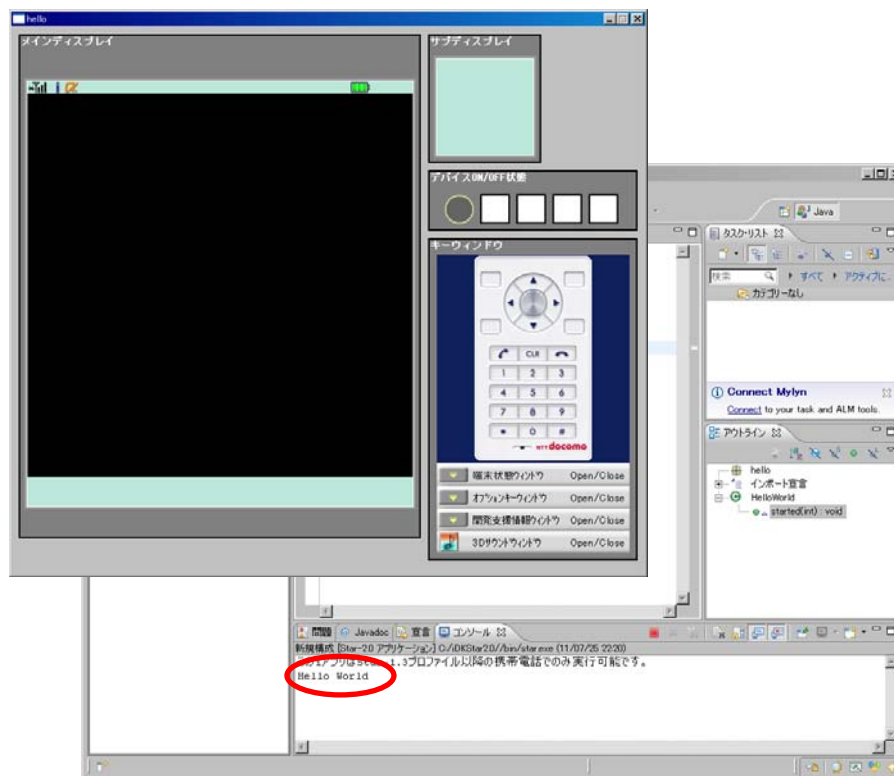


図 19-19 コンソールに出力された内容の表示

※ Eclipse プラグインについては、ネイティブアプリケーション表示・連携起動機能（4.16 メーラ起動、4.17 ブラウザ起動、4.18 コードリーダー起動、4.23 トルカ起動、4.25 BML起動、4.26 DTV起動、4.34 MyFACE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・DTV起動・MyFACE起動につきましては、通常起動と同様に「実行」ボタンを押した後即エミュレータが起動します。（起動タイプは「起動種別」で選択した種別が反映されます。）

19.3.8. プロジェクトの設定

プロジェクトの設定をするには：

1. 「プロジェクト」→「Star-2.0」→「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 19-20）。

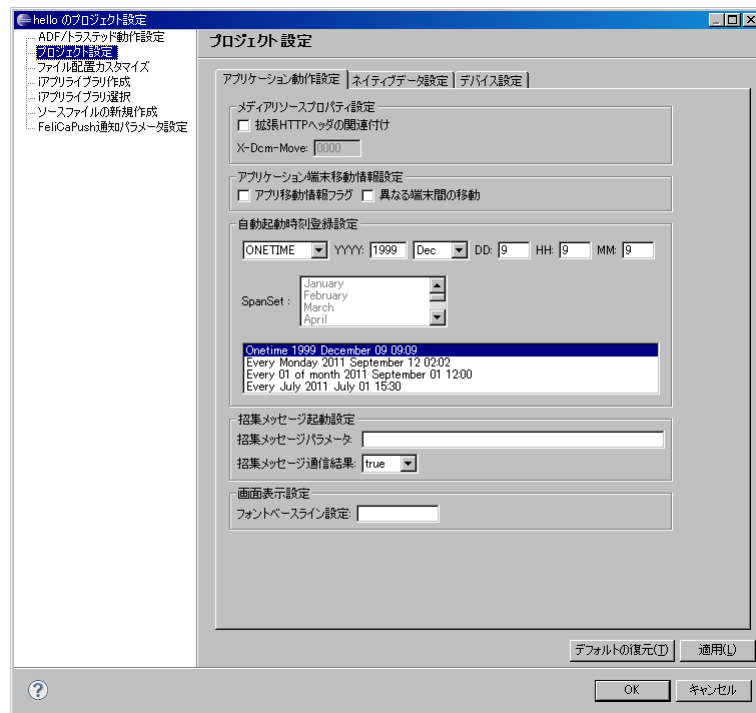


図 19-20 プロジェクト設定画面

各プロジェクト設定の設定項目については、「4.6. プロジェクトの設定」を参照してください。

19.3.9. 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：

1. 「プロジェクト」→「Star-2.0」→「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 19-21）。

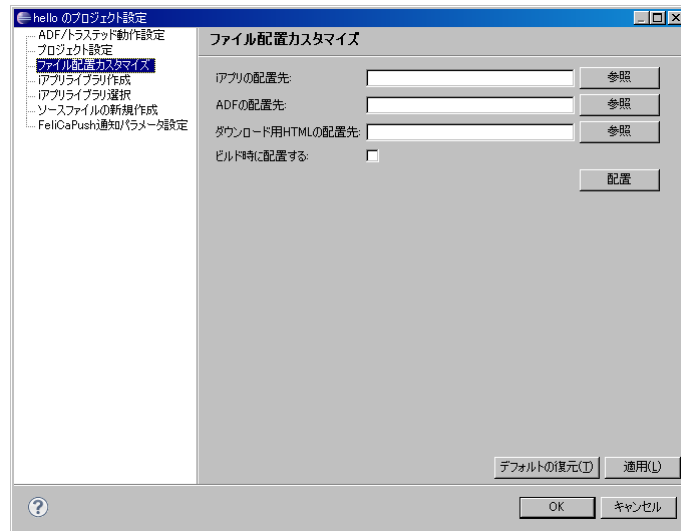


図 19-21 ファイル配置カスタマイズ

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、「4.7. ファイル配置カスタマイズ」を参照してください。

19.3.10. 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：

1. 「プロジェクト」→「Star-2.0」→「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 19-22）。

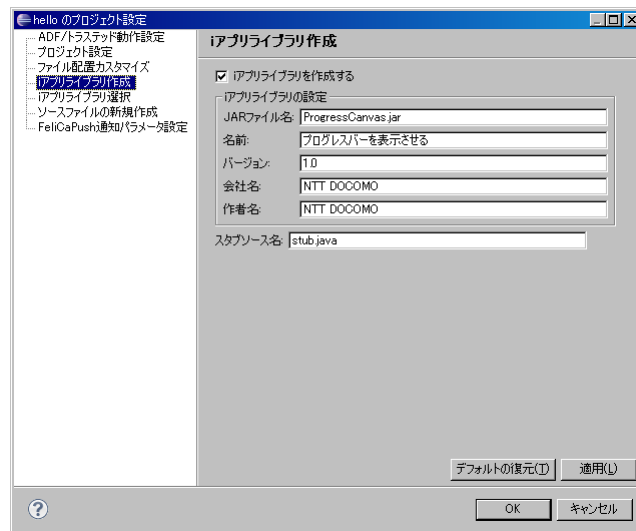


図 19-22 i アプリライブラリ作成

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、「9.7.1. i アプリライブラリの作成」を参照してください。

19.3.11. FeliCa Push 通知パラメータ設定

FeliCa Push の通知パラメータを設定するには：

1. 「プロジェクト」→「Star-2.0」→「FeliCa Push の通知パラメータ設定」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 19-23）。

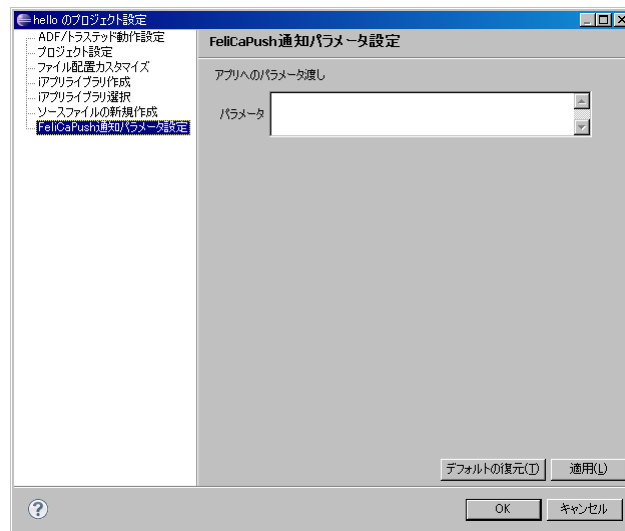


図 19-23 FeliCa Push 通知パラメータ設定

3. 「アプリへのパラメータ渡し」にパラメータを設定し、「OK」を押します。

詳細については、「8.10. FeliCaPush通知パラメータ設定」を参照してください。

19.3.12. 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. 「プロジェクト」→「Star-2.0」→「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 19-24）。

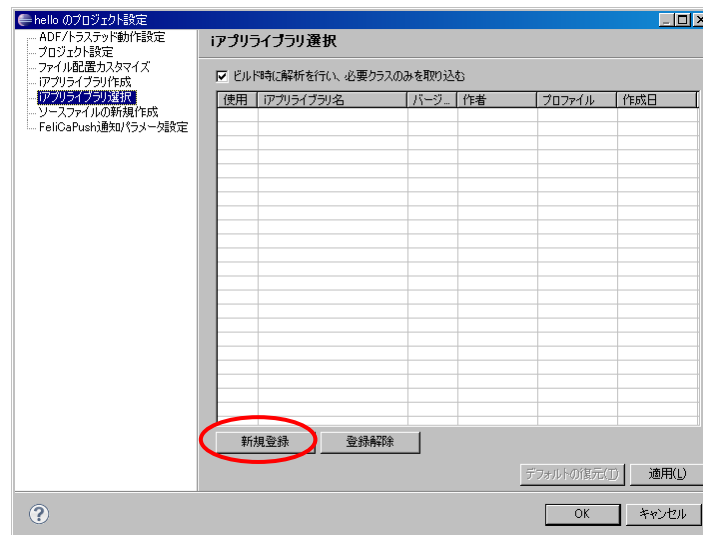


図 19-24 i アプリライブラリ選択

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 ファイル) 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 19-25）。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。

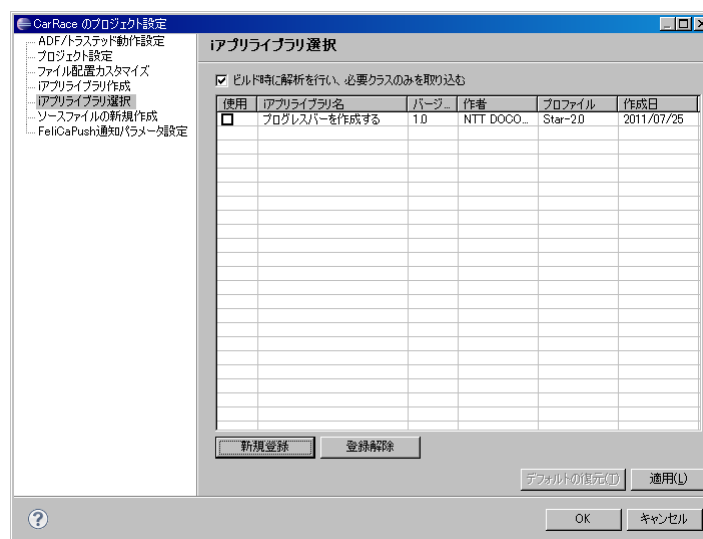


図 19-25 i アプリライブラリの選択画面 - ライブラリの追加

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:\¥iDKStar2.0¥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。

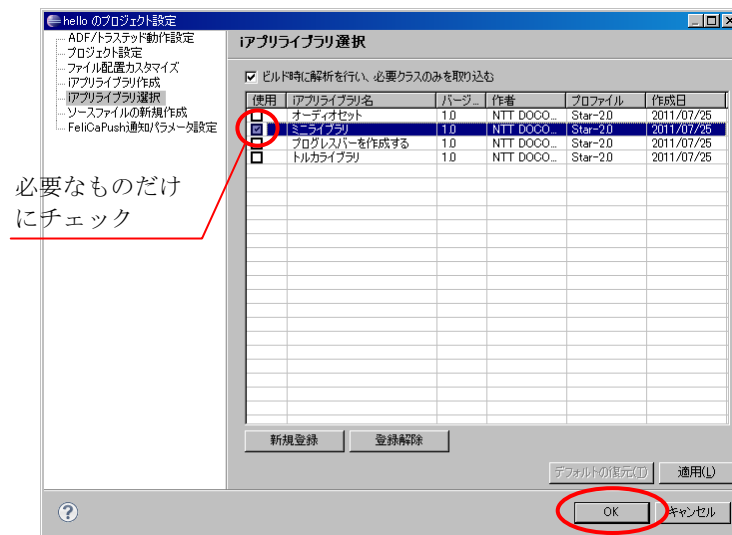


図 19-26 ライブラリの選択

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。
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、「9.7.2. i アプリライブラリの利用」を参照してください。

19.3.13. デバッグの実行

Eclipse での i アプリのデバッグは、フルアプリ、フル+ミニアプリ（フルアプリ実行時）についてのみ実行できます。

アプリケーションをデバッグする場合の例を以下に説明します。

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。

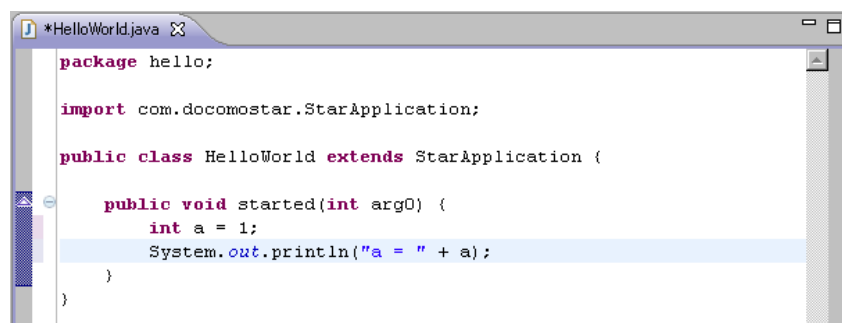


図 19-27 変数の表示

2. デバッグで止めたい行（ブレークポイント）の左枠をダブルクリックしてチェックを付けます（図 19-28）。

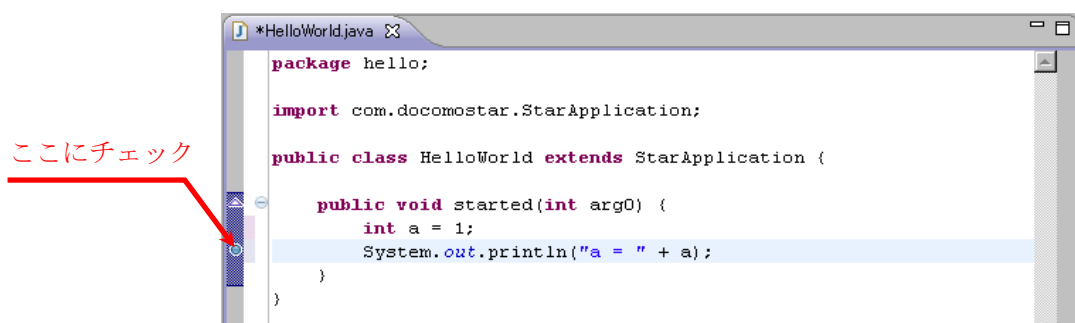


図 19-28 ブレークポイントの設定

3. 「実行」 → 「実行構成...」を押します。
4. 実行構成ウィンドウが表示されるので、実行構成の「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、「デバッグモード起動」のチェックを外して下さい（フルアプリ、フル+ミニアプリのみ実行可）。

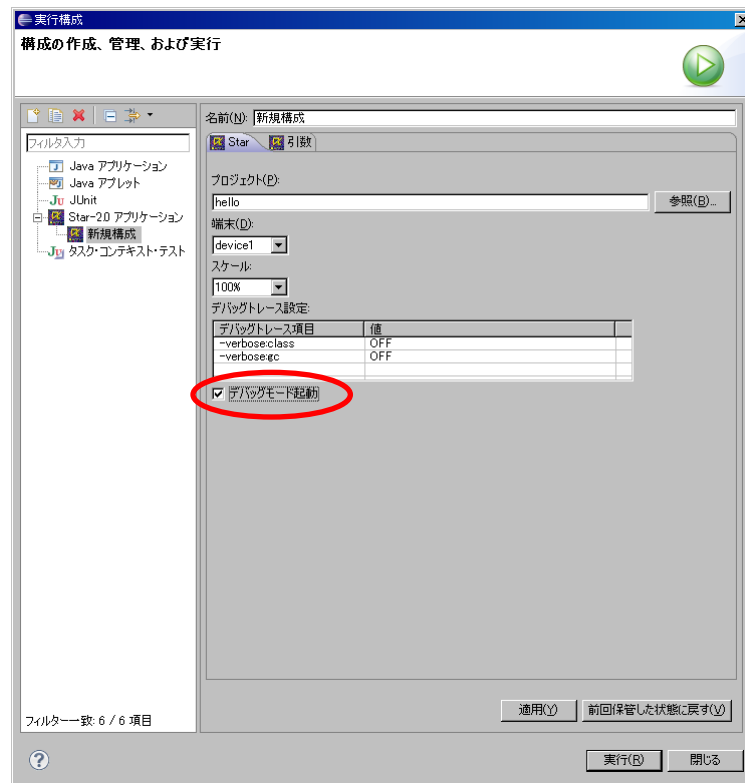


図 19-29 デバッグモードでの実行

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 19-30）。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。

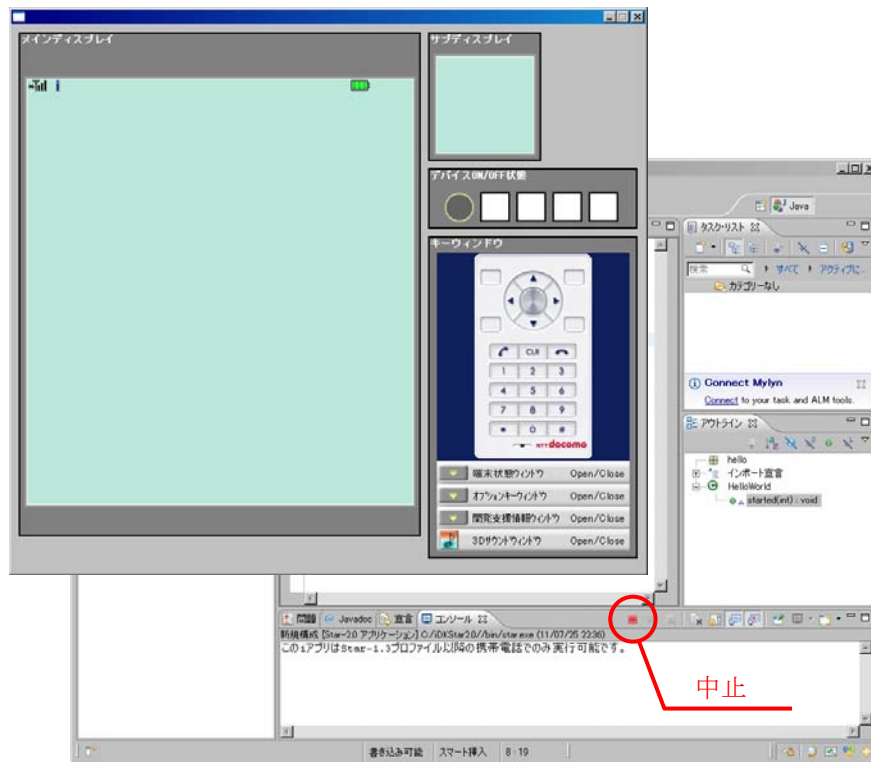


図 19-30 デバッグ実行 待ち受け状態

6. 次に「実行」 → 「**デバッグの構成...**」を選択します。
7. デバッグ構成ウィンドウが表示されるので、デバッグ構成の「**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、新規作成ボタンを押します。
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事ができます。
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 19-31）。
（この値は、一度設定するとそのまま保持されます。）

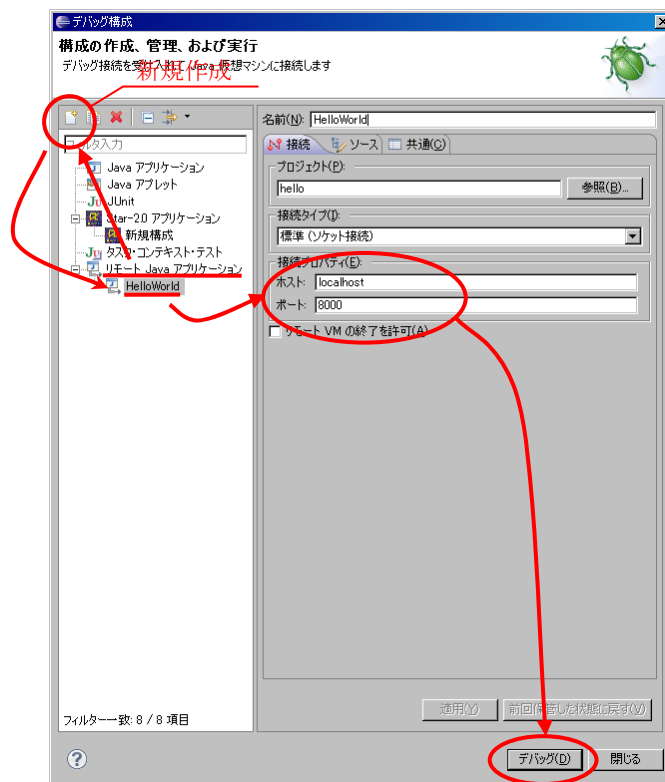


図 19-31 デバッグ接続画面

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 19-32）。

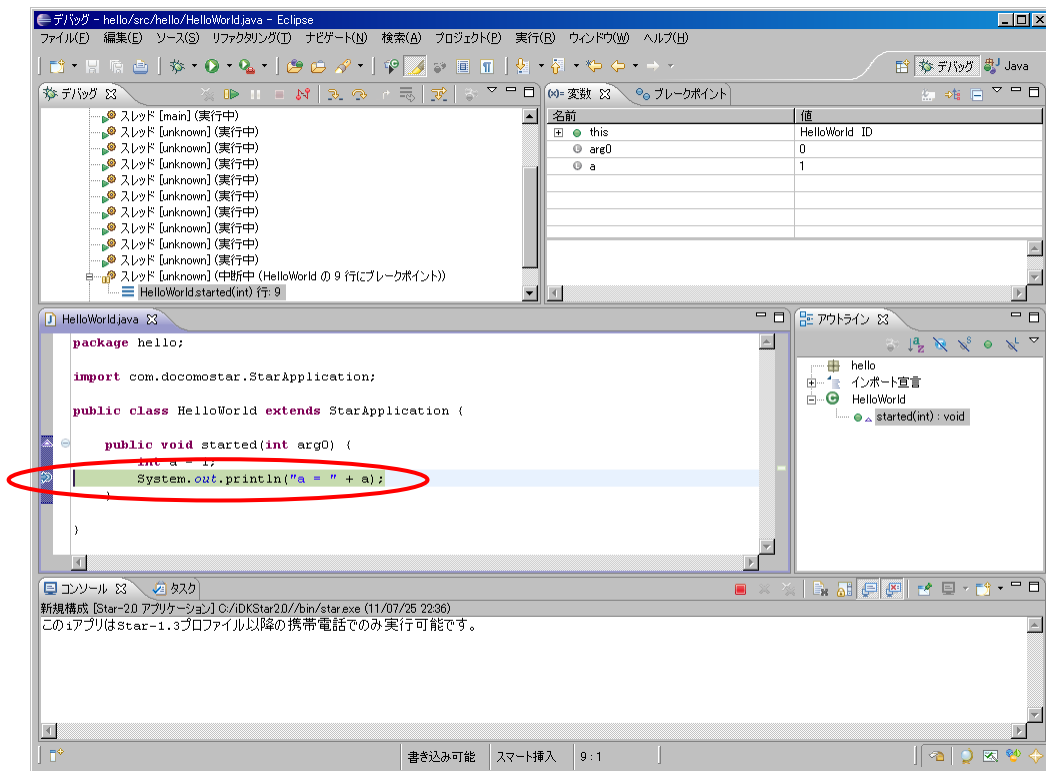


図 19-32 ブレークポイントでの処理の中断

11. 変数ウィンドウのウォッチポイントで、「a」の行を右クリックし、「値の変更」(もしくは「変数の変更」)を選択します(図 19-33)。

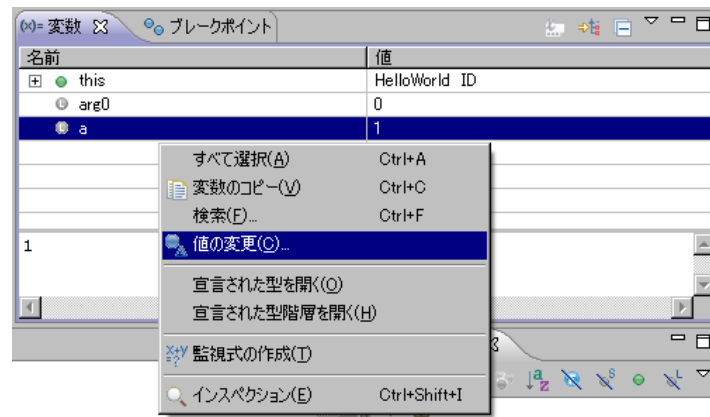


図 19-33 変数の変更

12. 「プリミティブ値の変更」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 19-34)。

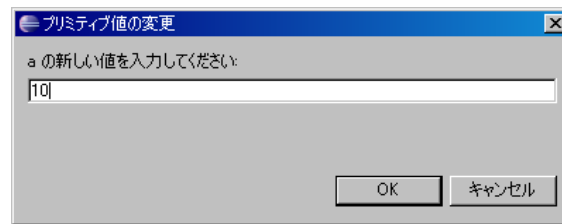


図 19-34 変数値の設定

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。
変数を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。

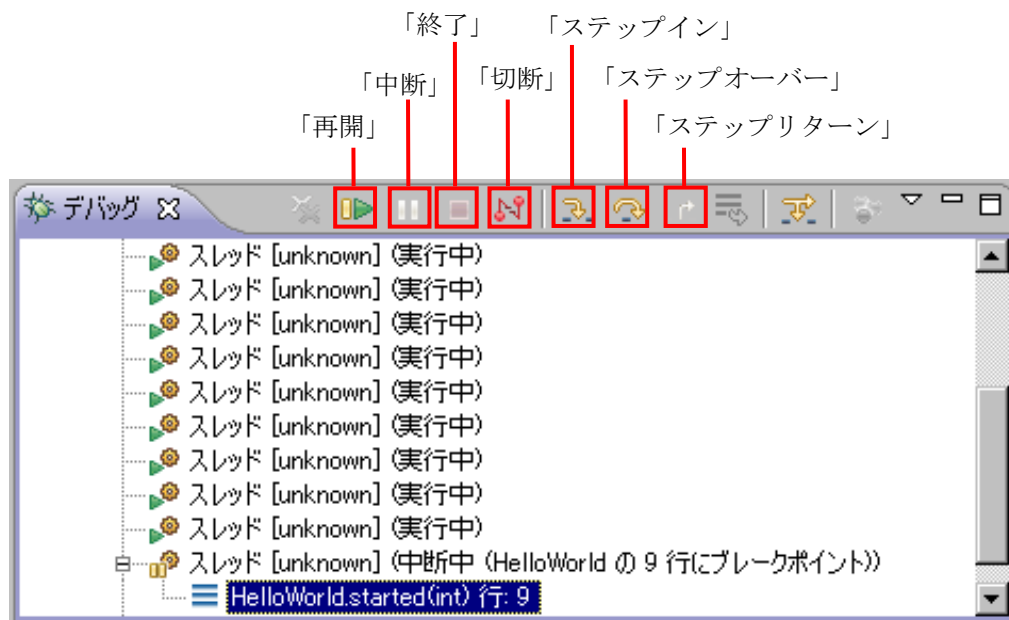


図 19-35 デバッグビューのボタン

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。

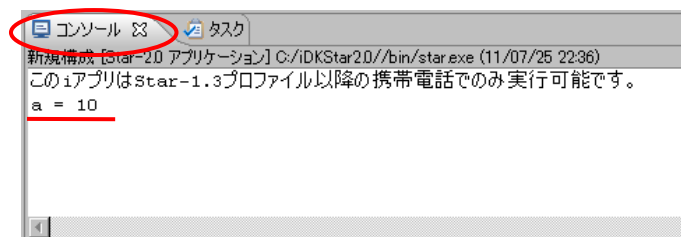


図 19-36 デバッグ実行結果

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。

19.3.14. 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 for iα ppli」には「iα ppliTool」の DTV画面（図 4-56）に相当する画面が存在しないため、Eclipseから起動した i アプリからDTVを連携起動する場合は以下のような流れとなります（図 19-37）。

1. i アプリ上で DTV 起動（"Launcher.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す事で i アプリに復帰（レジューム）します（③）。

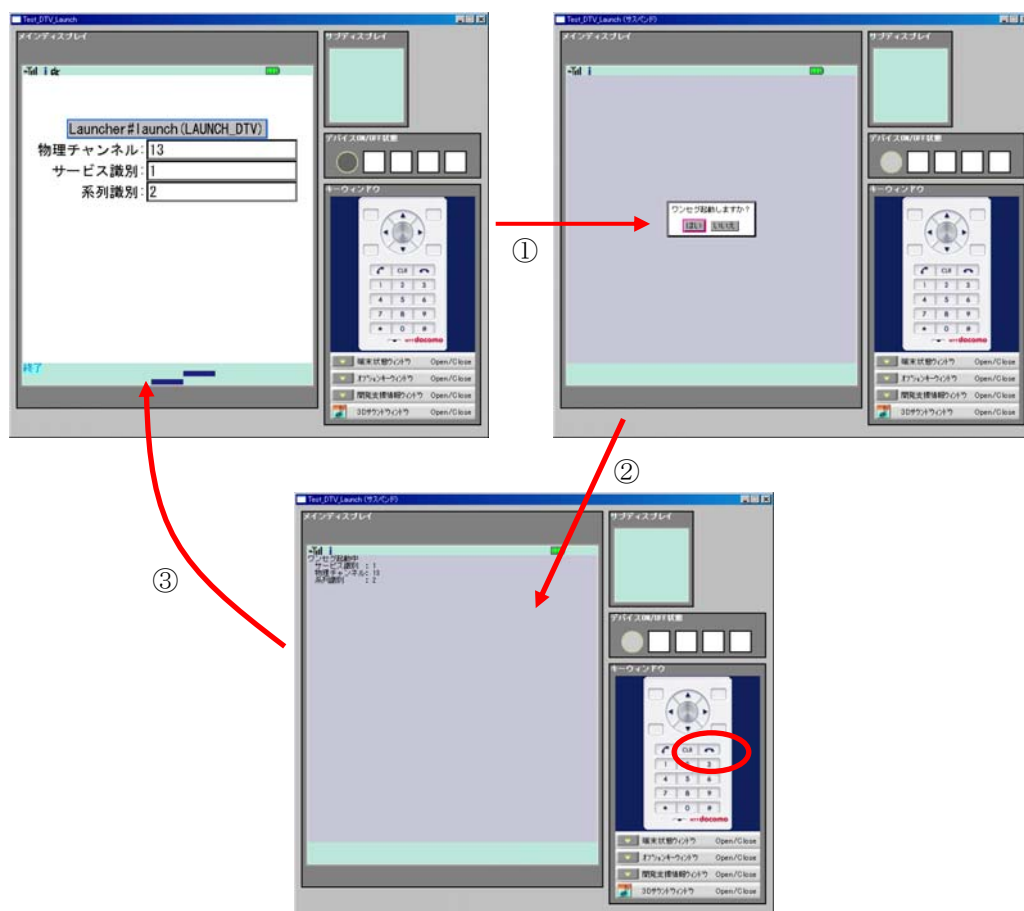


図 19-37 Eclipse から起動した i アプリ上での DTV 起動

19.4. 各種設定

19.4.1. 全般設定

i モード携帯電話上での全体の機能に関連する値を設定することができます。
ここでは、以下の項目を設定することが可能です。

- ・ ヒープサイズ
- ・ タイムゾーン
- ・ 対応フォントサイズ
- ・ 外部メモリ
- ・ 内部ストレージ

アプリケーション動作全般の設定をするには：

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、「Star-2.0 Environment」を選択します
(図 19-38)。

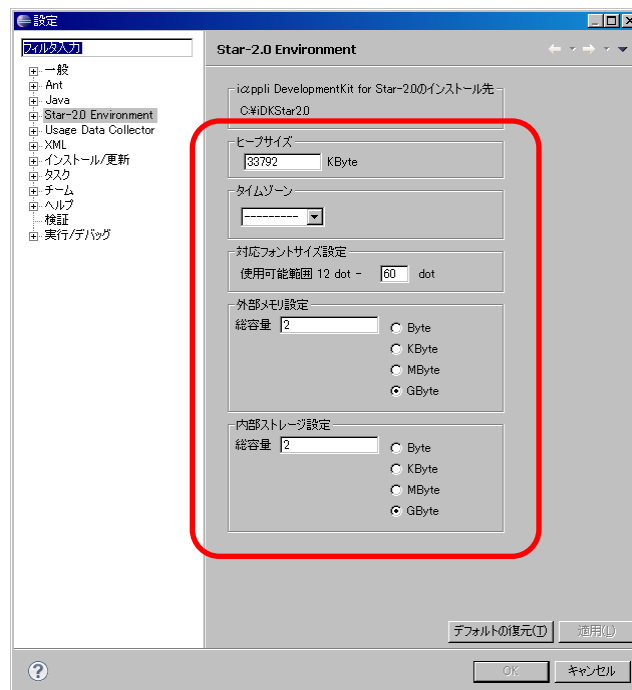


図 19-38 Eclipse 設定ウィンドウ(全般)

3. それぞれ必要な値を設定して「OK」を押します。

各設定項目の詳細については、「6.2. 全般設定」を参照してください。

19.4.2. 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：

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 「STAR_CLASSES/lib/classes.zip」ライブラリを選択して「除去」を押します。（もし、STAR_CLASS_ZIP になっていれば必要ありません。）
 - ③ 同様に「STAR_CLASSES/lib/star_classes.zip」ライブラリを選択して「除去」を押します。（もし、STAR_STARCLASSES_ZIP になっていれば必要ありません。）

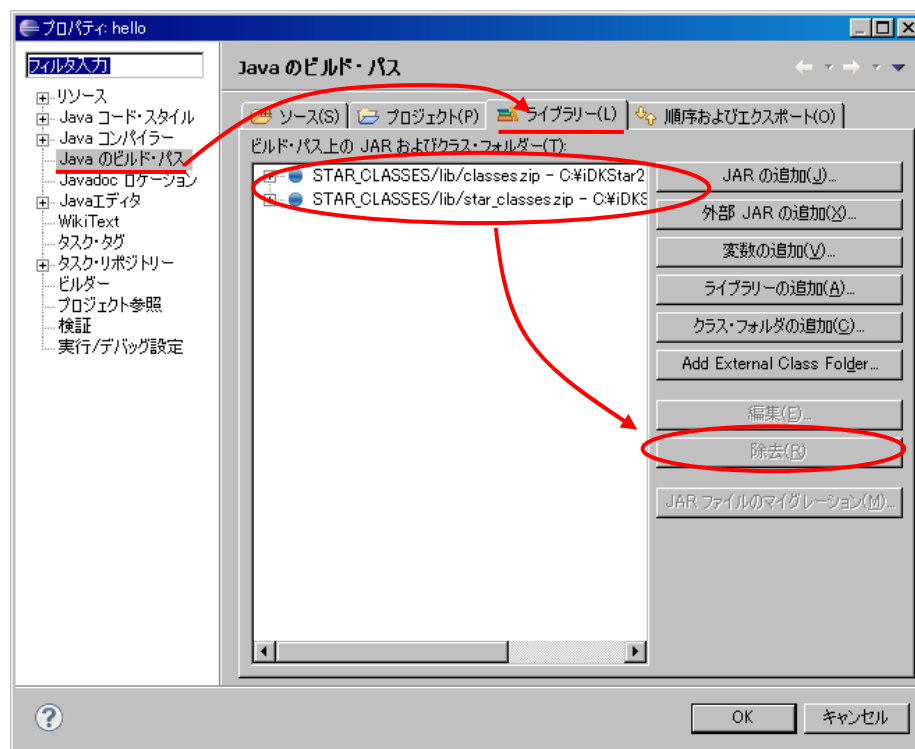


図 19-39 Java ライブラリ設定ウィンドウ

- ④ 「変数の追加」を押します。

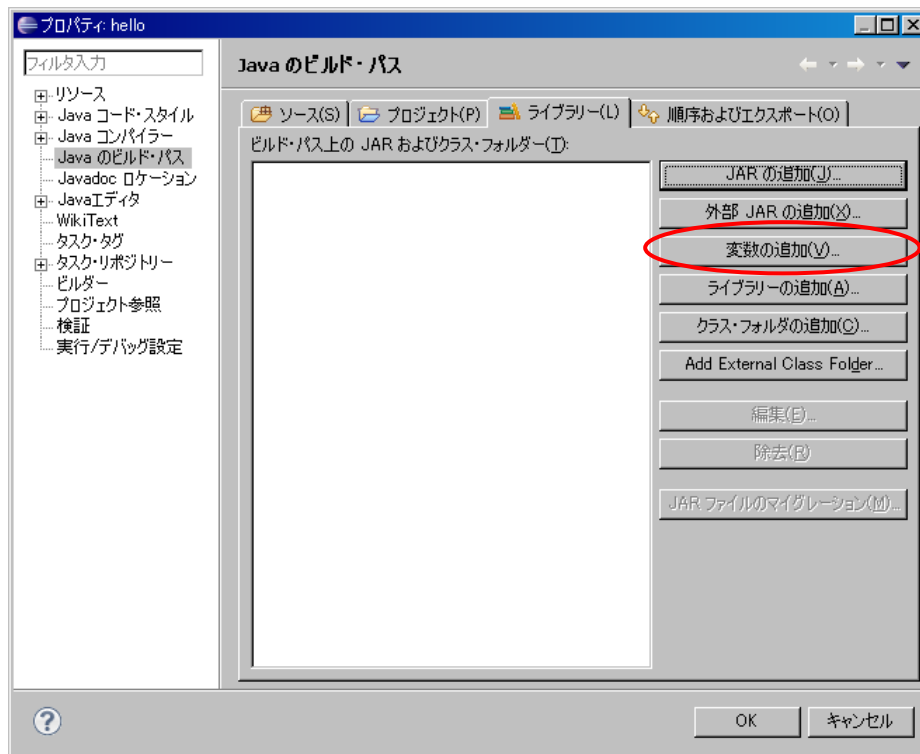


図 19-40 Java ライブラリー設定ウィンドウ（変数の追加）

- ⑤ 変数一覧から「STAR_CLASS_ZIP」と「STAR_STARCLASS_ZIP」を選択して「OK」を押します。

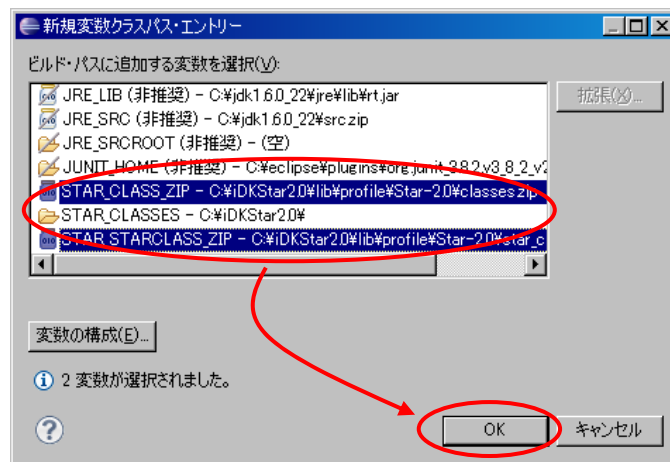


図 19-41 新規変数クラスパス・エンタリーウィンドウ

- ⑥ 一覧に選択した変数（「STAR_CLASS_ZIP」と「STAR_STARCLASS_ZIP」）が追加されていることを確認して「OK」を押します。

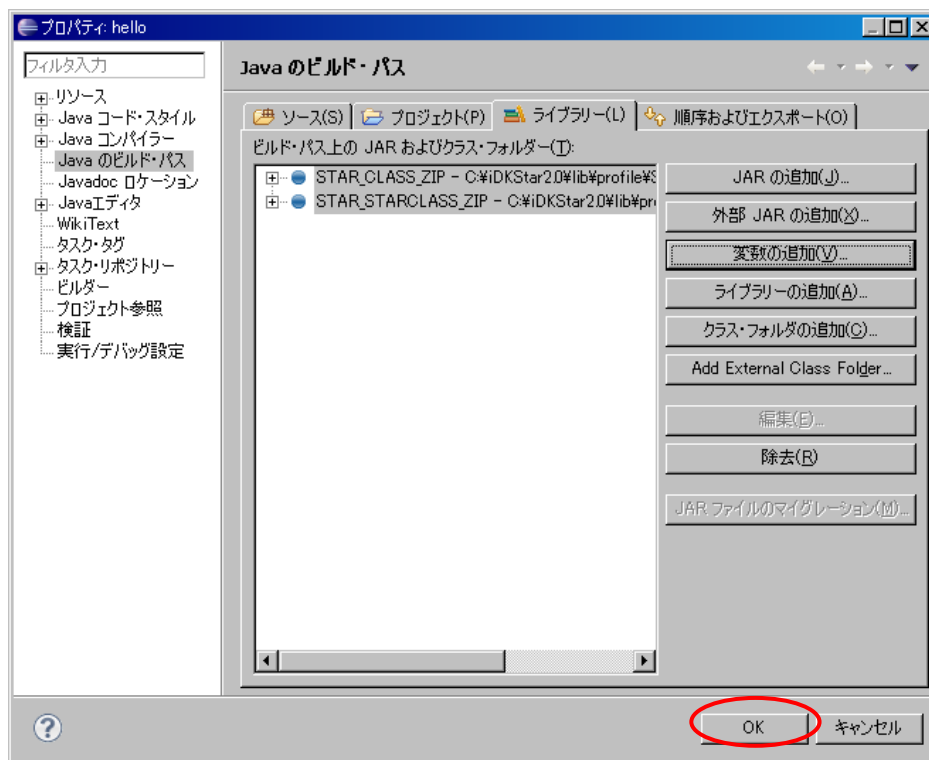


図 19-42 Java ライブラリー設定ウィンドウ（再設定後）

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、「Star-2.0 Environment」→「ビルド設定」を選択します（図 19-43）。

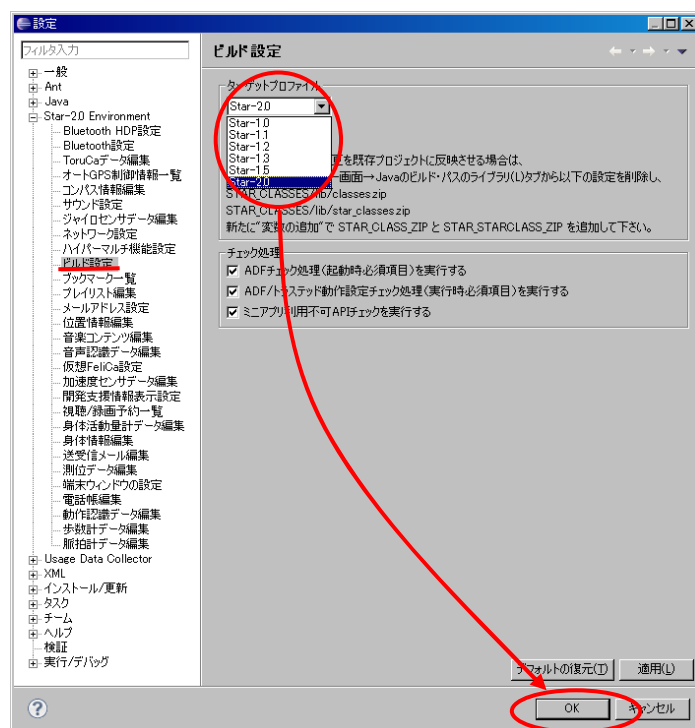


図 19-43 ビルド設定

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。

※ ターゲットプロファイル設定にかかわらず、「i アプリ Development Kit for Star-2.0」でビルドした i アプリ (JAR ファイル) は、Star-1.3 プロファイル搭載以降の携帯電話でのみ実行可能です。それ以前の携帯電話では実行できませんのでご注意ください。

詳細については、「8.1.2. ターゲットプロファイルの設定」を参照してください。

19.4.3. ADF/トラステッド動作設定チェック処理の設定

ADF/トラステッド動作設定チェック処理の設定をするには：

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、「Star-2.0 Environment」→「ビルド設定」を選択します（図 19-44）。



図 19-44 ビルド設定

3. 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 **API 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。
ミニアプリのビルド時に **ミニアプリで利用できない API を使っていた場合** にビルドを中断したい時は、「ミニアプリ利用不可 API チェック処理を実行する」にチェックを付けます。
4. チェック処理の設定を変更して「OK」を押します。
これにより、i アプリの起動前に実施されるビルド時に必要なAPIを構文解析して警告を表示します（図 19-45）。

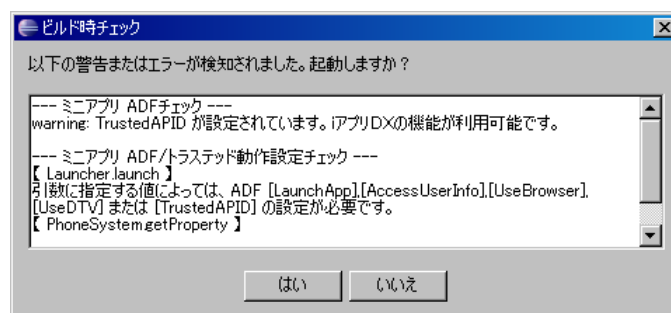


図 19-45 ADF/トラステッド動作設定のチェック結果

19.4.4. ネットワークの設定

ネットワークの設定をするには：

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、「Star-2.0 Environment」→「ネットワーク設定」を選択します（図 19-46）。



図 19-46 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、「6.1. ネットワークの設定」を参照してください。

19.4.5. メールアドレスの設定

メールアドレスの設定をするには：

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、「Star-2.0 Environment」→「メールアドレス設定」を選択します（図 19-47）。

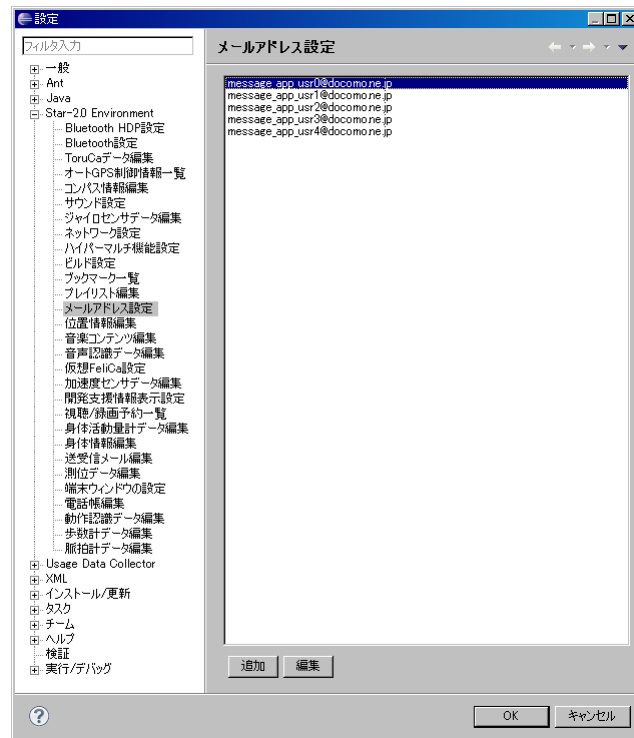


図 19-47 メールアドレス設定画面

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 19-48）。



図 19-48 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。

19.4.6. 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ウィンドウの設定をするには：

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、「Star-2.0 Environment」→「端末ウィンドウの設定」を選択します（図 19-49）。

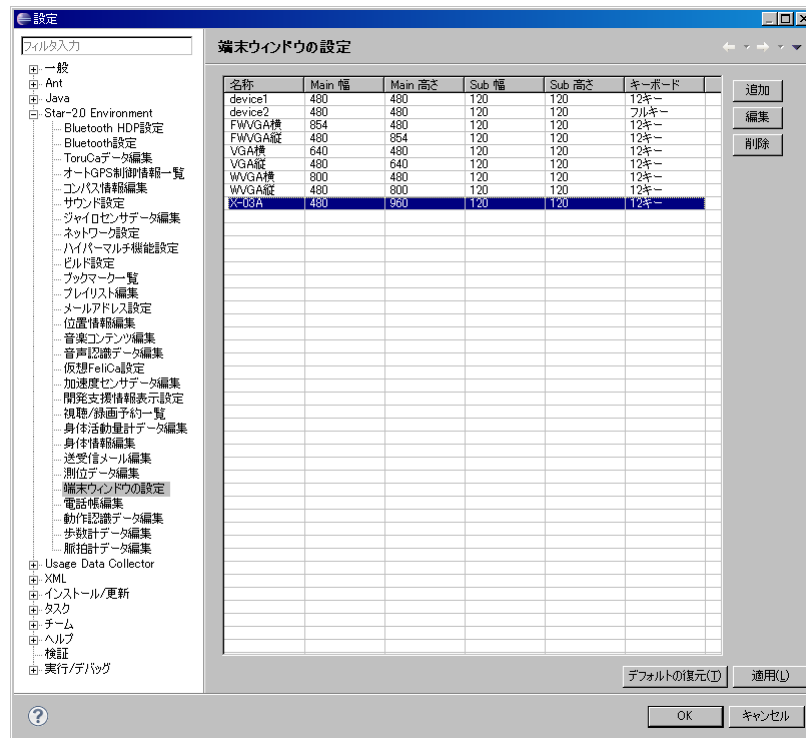


図 19-49 端末ウィンドウの設定画面

3. 端末ウィンドウの設定を追加するには「追加」、変更するには変更したい端末ウィンドウの設定を選択し、「編集」を押します。
端末ウィンドウ編集画面が表示されます（図 19-50）。

端末ウィンドウ設定

名称

メインディスプレイ

幅

高さ

サブディスプレイ

幅

高さ

キーボード

立体視

解像度比

深さ0の視差オフセットの値

深さが1変化する時に視差オフセットが変化する値

立体視強度

オプション機能対応設定

設定 取消

図 19-50 端末ウィンドウ編集画面

4. 端末ウィンドウの設定値を入力してください。
5. オプション機能の対応／非対応を設定するには「オプション機能対応設定」を押します。
オプション機能対応設定画面が表示され、選択した端末ウィンドウ設定についての現在のオプション機能対応設定状態が表示されます。（図 19-51）
初期状態では、すべて「対応」に設定されてますので、非対応としたい機能について「非対応」を選択してください。
「設定」を押すと、端末ウィンドウ編集画面に戻ります。

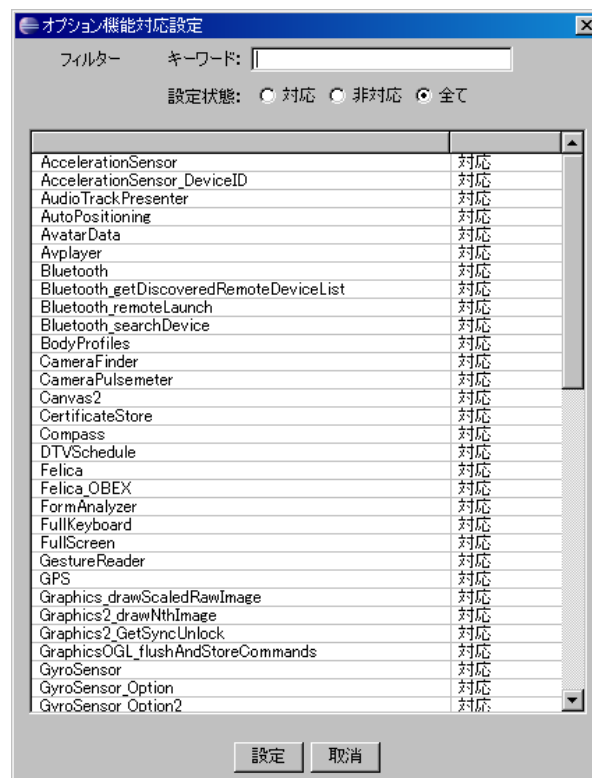


図 19-51 オプション機能対応設定画面

6. 「設定」を押すと、端末ウィンドウの設定画面に戻ります。
端末ウィンドウの設定画面の一覧に追加・変更が反映されます。
7. 端末ウィンドウ設定を削除したい場合は、一覧から端末ウィンドウの設定を選択し、「削除」を押します。

各設定項目の詳細については、「8.6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。

19.4.7. 開発支援情報表示設定

開発支援情報画面の設定をするには：

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、「Star-2.0 Environment」→「開発支援情報表示設定」を選択します（図 19-52）。



図 19-52 開発支援情報表示設定画面

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、「8.9 開発支援情報表示設定」を参照してください。

19.4.8. 電話帳の編集

電話帳の編集をするには：

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、「Star-2.0 Environment」→「電話帳編集」を選択します（図 19-53）。

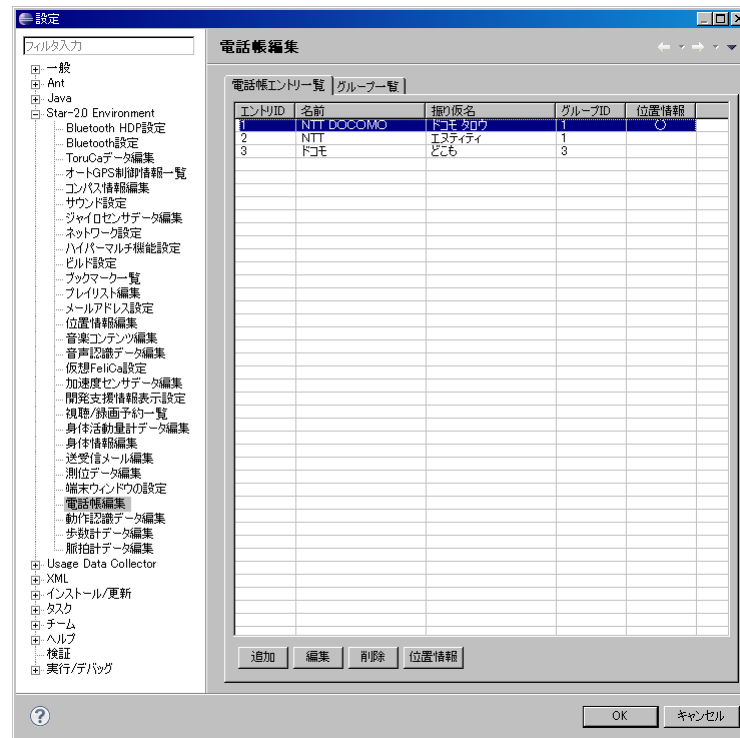


図 19-53 電話帳編集画面

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 19-54）。

電話帳エントリ登録

名前 姓 NTT
名 DOCOMO
振仮名姓 ドコモ
名 タロウ
電話番号 1 090xxxxxxx
2 03xxxxxxx
3
メールアドレス 1 message_app_usr1@
2
3
グループID T1

設定 取消

図 19-54 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 7-1. 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 19-55）。

位置情報編集

位置測定日時
2010 / 01 / 01 12 : 34 : 56 . 789
測定レベル
☐ なし ☐ 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 削除 取消

図 19-55 電話帳位置情報編集画面

19.4.9. グループの編集

グループの編集をするには：

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、「Star-2.0 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 19-56）。

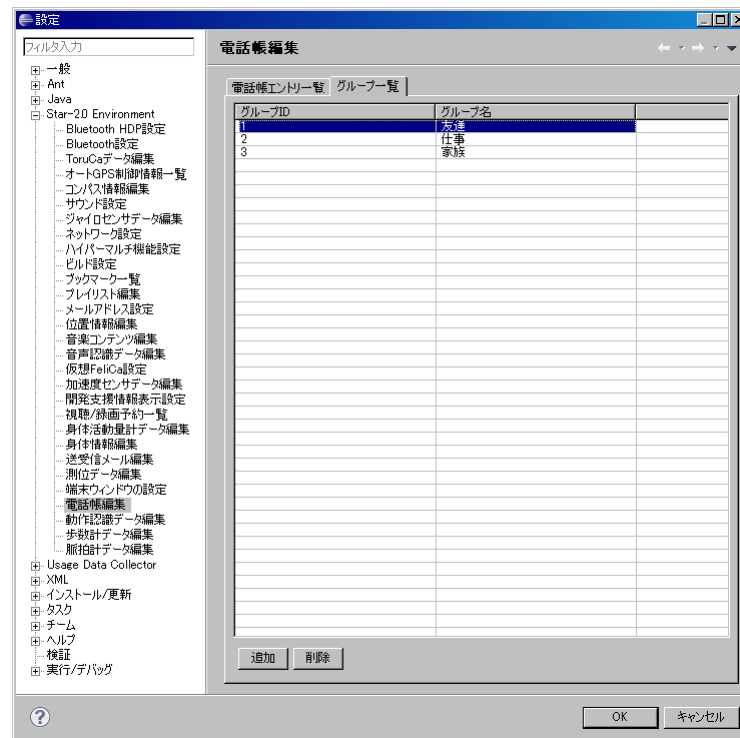


図 19-56 Eclipse 設定ウィンドウのグループ編集画面

3. グループを追加したい場合は「追加」を押します（図 19-57）。

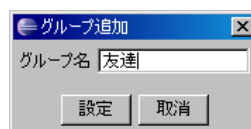


図 19-57 グループ追加画面

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。

19.4.10. 送受信メールの編集

送受信メールの編集をするには：

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、「Star-2.0 Environment」→「送受信メール編集」を選択します（図 19-58）。



図 19-58 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 19-59、図 19-60)。

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: 2009 年 07 月 17 日 15 時 58 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 19-59 受信メール編集画面

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: 2009 年 07 月 17 日 16 時 01 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 19-60 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、「7.4. 送受信メールの編集」を参照してください。

19.4.11. ブックマーク一覧

ブックマーク一覧を表示するには:

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、「Star-2.0 Environment」→「ブックマーク一覧」を選択します（図 19-61）。

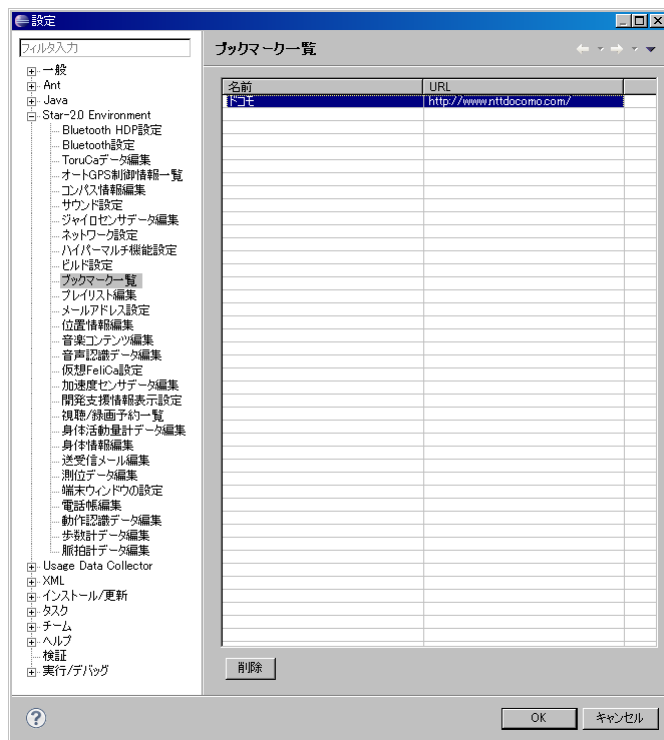


図 19-61 ブックマーク一覧画面

- 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。

19.4.12. サウンド設定

サウンドの設定をするには：

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、「Star-2.0 Environment」→「サウンド設定」を選択します（図 19-62）。



図 19-62. 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、「6.4. サウンド設定」を参照してください。

19.4.13. 位置情報編集

位置情報の編集をするには：

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、「Star-2.0 Environment」→「位置情報編集」を選択します（図 19-63）。

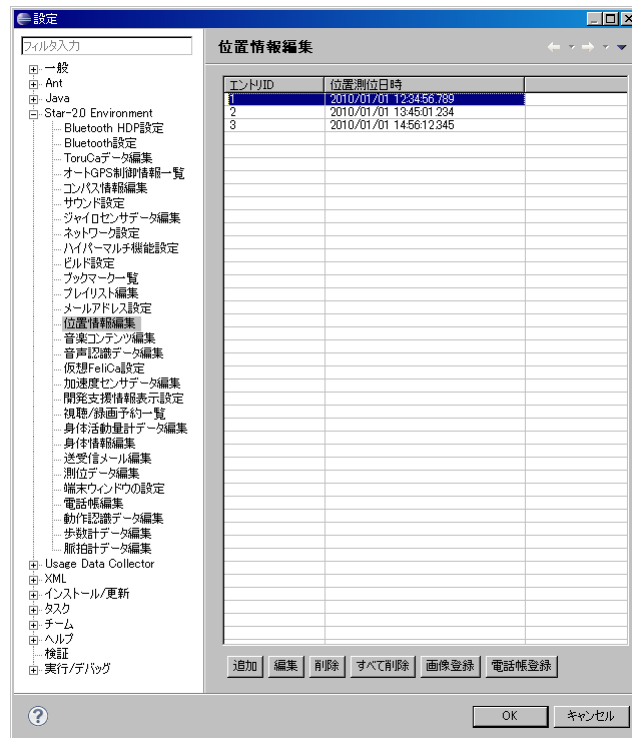


図 19-63 Eclipse 設定ウィンドウの位置情報編集一覧画面

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 19-64）。

位置情報編集

エントリID 3

位置測位日時
2010 / 01 / 01 14 : 56 : 12 345

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 19-64 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、「7.5. 位置情報編集」を参照してください。

19.4.14. 測位データ編集

測位データの編集をするには：

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、「Star-2.0 Environment」→「測位データ編集」を選択します（図 19-65）。



図 19-65 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。

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
5 秒

測地系 WGS84

測位方式 GPS測位

設定 取消

図 19-66 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、「7.6. 測位データ編集」を参照してください。

19.4.15. コンパス情報編集

電子コンパスの情報の編集をするには：

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、「Star-2.0 Environment」→「コンパス情報編集」を選択します（図 19-67）。

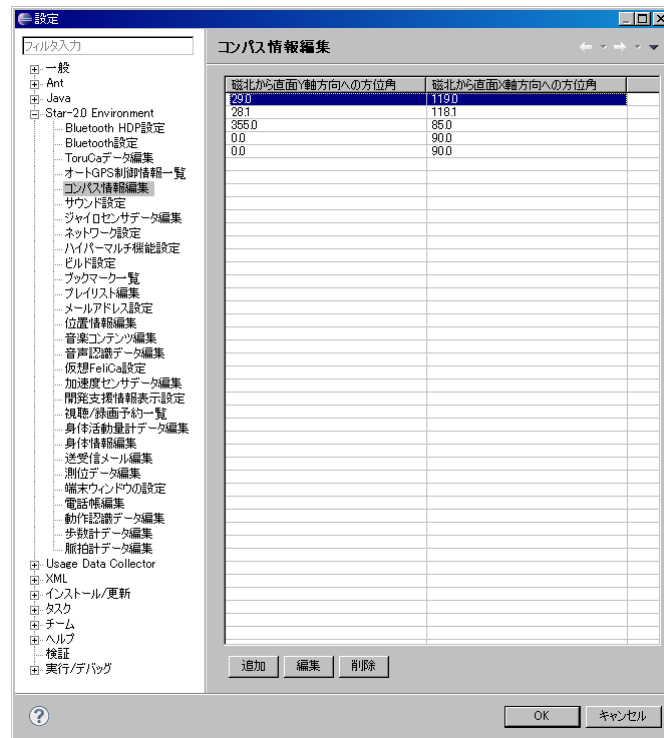


図 19-67 Eclipse 設定ウィンドウのコンパス情報編集画面

3. 電子コンパスデータを追加したい場合は「追加」、電子コンパスデータを編集したい場合は一覧から編集したい電子コンパスデータを選択して、「編集」ボタンを押します（図 19-68）。

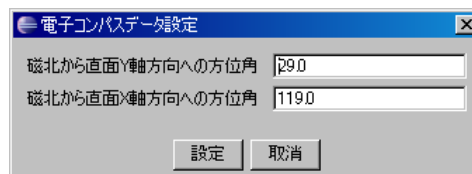


図 19-68 測位データ編集画面

4. 各設定項目を設定します。
5. 「設定」を押すと、コンパス情報編集画面に追加されます。
6. 電子コンパスデータを削除したい場合は、一覧から削除したい電子コンパスデータを選択して「削除」ボタンを押します。

各設定項目の詳細については、「7.7. コンパス情報編集」を参照してください。

19.4.16. ToruCa データ編集

トルカデータの編集をするには：

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、「Star-2.0 Environment」→「ToruCaデータ編集」を選択します（図 19-69）。

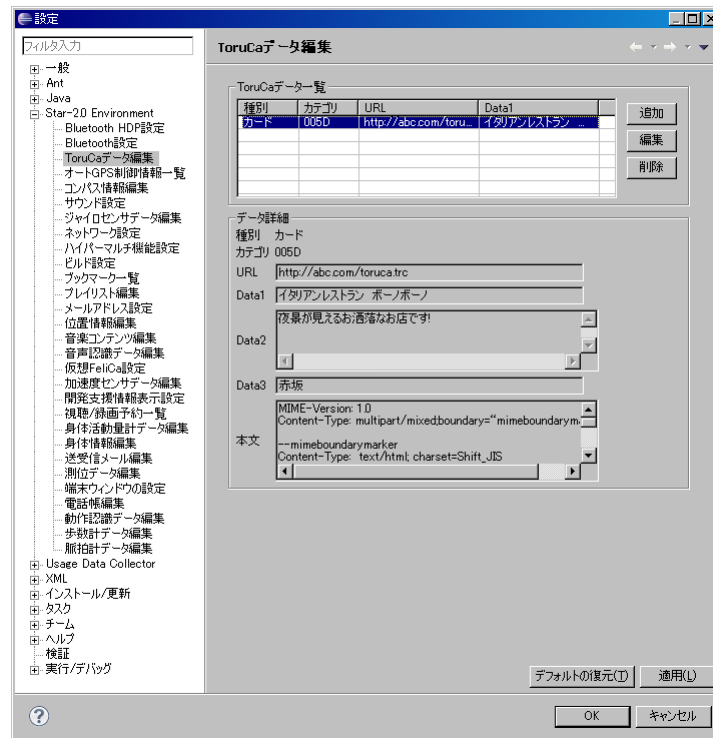


図 19-69 Eclipse 設定画面の ToruCa データ編集一覧画面

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 19-70）。

ToruCa編集画面

データ詳細

種別: カード

ジャンル: ファッション/アクセサリ/雑貨/インテリア

コード: 005D

バージョン: 2.0

IP-ID: 12345678

カナID: ポー

再配布識別子: 4

色ID: 9

有効期限: 051225

URL: http://abc.com/toruca.trc

Data1: イタリアンレストラン ポーノポーノ

Data2: 夜景が見えるお洒落なお店です!

Data3: 赤坂

本文: MIME-Version: 1.0
Content-Type: multipart/mixed;boundary="mimeboundarymarker"
--mimeboundarymarker
Content-Type: text/html; charset=Shift_JIS
Content-Transfer-Encoding: 8bit
(本文内容)
--mimeboundarymarker

設定 取消

図 19-70 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、「7.8. トルカデータ編集」を参照してください。

19.4.17. Bluetooth 設定

Bluetooth の設定をするには：

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、「Star-2.0 Environment」→「Bluetooth設定」を選択します（図 19-71）。

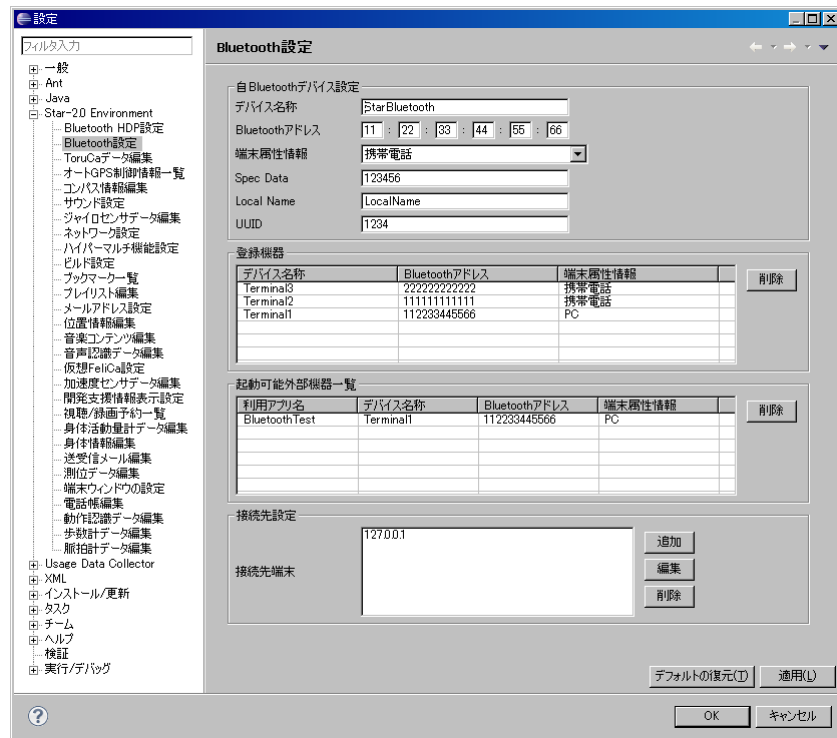


図 19-71. Bluetooth 設定画面

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. 起動可能登録機器を削除したい場合は、一覧から選択して「削除」を押してください。
6. 接続先端末の探索先ホスト名を追加したい場合は「追加」、ホスト名を変更・削除したい場合は一覧からホスト名を選択して「編集」「削除」を押してください。
7. 「OK」を押すと、設定されます。

各設定項目の詳細については、「6.6.1 Bluetooth設定」を参照してください。

19.4.18. Bluetooth HDP 設定

Bluetooth HDP 機器の設定をするには：

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、「Star-2.0 Environment」→「Bluetooth HDP 設定」を選択します（図 19-72）。

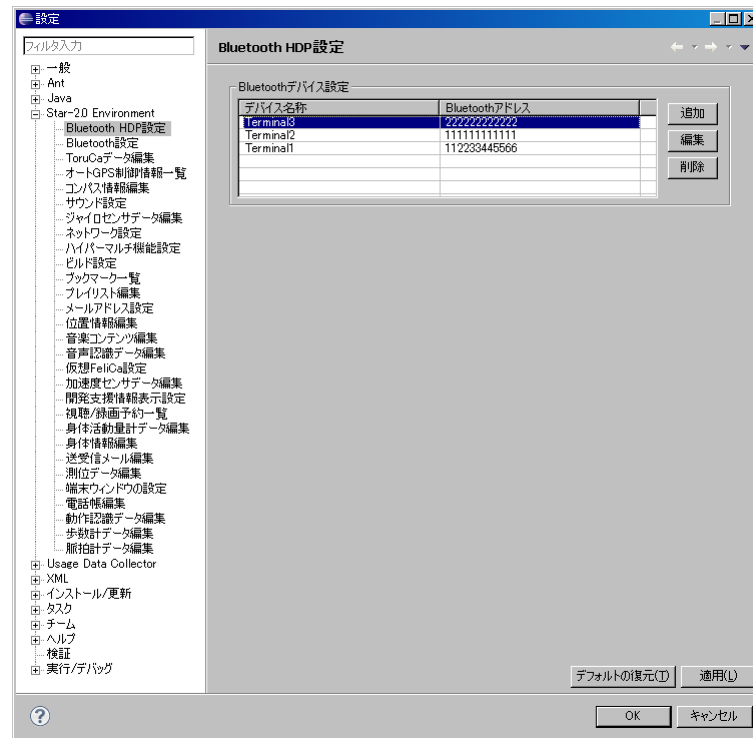


図 19-72. Bluetooth HDP 設定画面

3. HDP機器を追加するには「追加」、変更するには変更したいHDP機器を選択し、「編集」を押してください。
HDP設定画面が表示されます。（図 19-73）

HDP設定

デバイス名称: Terminal1

Bluetoothアドレス: 11 : 22 : 33 : 44 : 55 : 66

端末属性情報: PC

Spec Data: 1234

Local Name: Terminal1

UUID: 1234

MDEP登録一覧

ID	Type	Role
1	血圧計	Source

追加 編集 削除

設定 取消

図 19-73 HDP 設定画面

4. 各設定項目を設定します。
5. MDEPを追加したい場合は「追加」、MDEPを編集したい場合は一覧から編集したいMDEPデータを選択して、「編集」を押してください。（図 19-74）

MDEP設定

ID: 1

Type: 血圧計

Role: Source

設定 取消

図 19-74MDEP 設定画面

6. 各設定項目を設定して「設定」を押すと、HDP 設定画面に戻ります。
7. HDP 設定画面で「設定」を押すと、Bluetooth HDP 設定画面に戻ります。
8. HDP 機器情報を削除したい場合は、一覧から削除したい HDP 機器情報を選択して「削除」を押してください。

各設定項目の詳細については、「6.6.2 Bluetooth HDP設定」を参照してください。

19.4.19. 仮想 FeliCa 設定

仮想 FeliCa 動作の設定をするには：

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、「Star-2.0 Environment」→「仮想FeliCa設定」を選択します（図 19-75）。

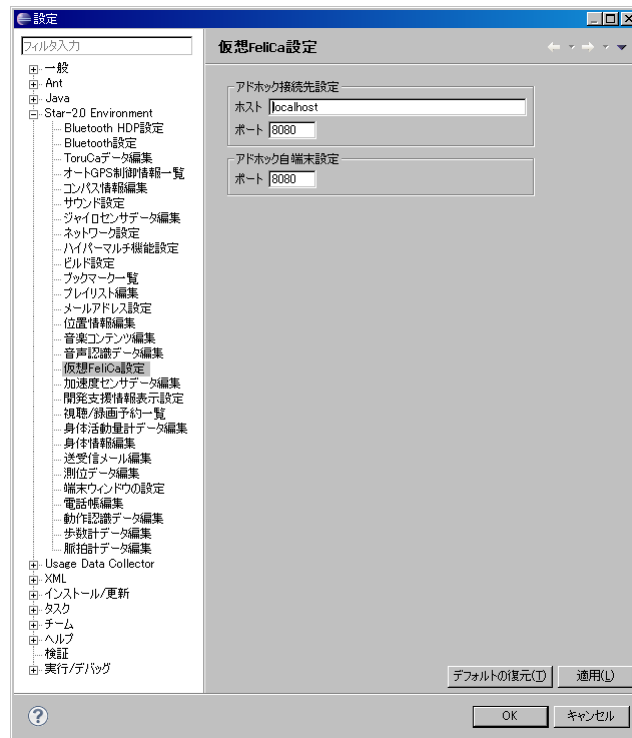


図 19-75. 仮想 FeliCa 設定画面

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、「6.7. 仮想FeliCa設定」を参照してください。

19.4.20. 視聴/録画予約一覧

視聴/録画予約一覧の表示をするには：

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、「Star-2.0 Environment」→「視聴/録画予約一覧」を選択します（図 19-76）。

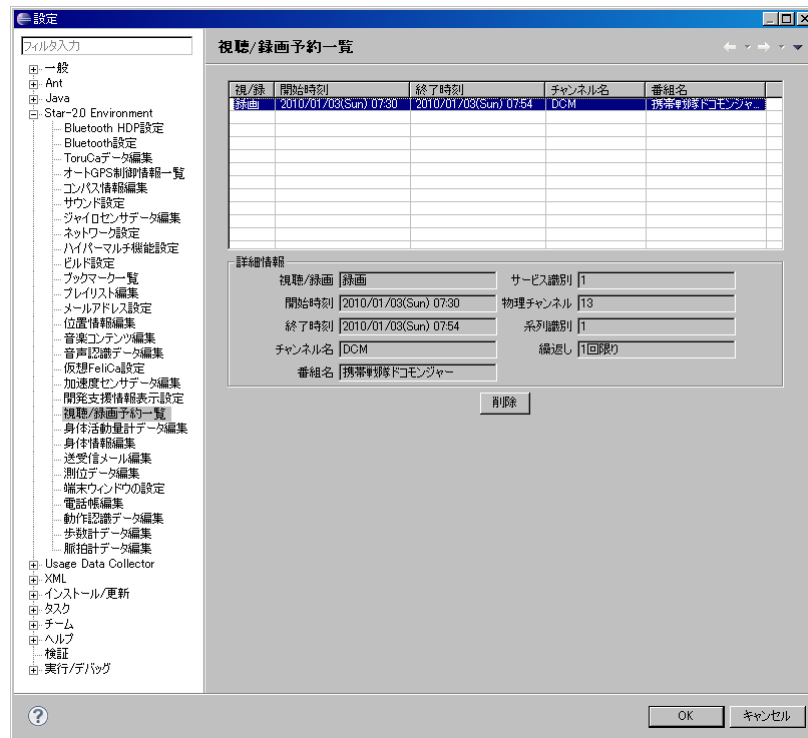


図 19-76. 視聴/録画予約一覧画面

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。

19.4.21. 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には：

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、「Star-2.0 Environment」→「ハイパーマルチ機能設定」を選択します（図 19-77）。



図 19-77 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、「6.8. ハイパーマルチ機能設定」を参照してください。

19.4.22. 加速度センサデータ編集

加速度センサデータを編集するには:

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、「Star-2.0 Environment」→「加速度センサーデータ編集」を選択します（図 19-78）。

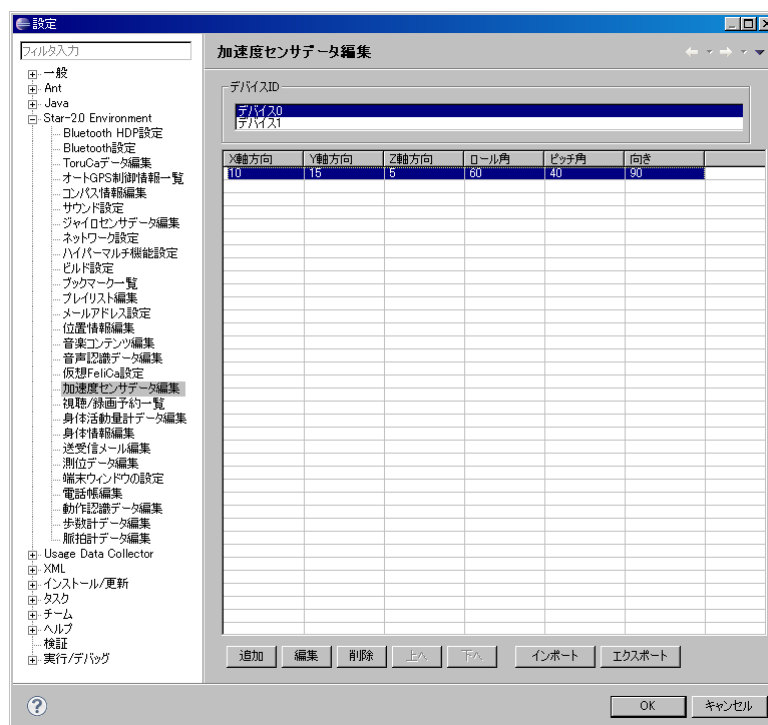


図 19-78 加速度センサデータ編集画面

- 編集するデバイス ID を選択します。
- 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 19-79)。



5. 各設定項目を設定します。
6. 「設定」を押すと、加速度センサデータ一覧に追加されます。
7. 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。
8. 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。
9. 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。
10. 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、「7.10. 加速度センサデータ編集」を参照してください

19.4.23. 動作認識データ編集

動作認識データを編集するには：

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、「Star-2.0 Environment」→「動作認識データ編集」を選択します（図 19-80）。

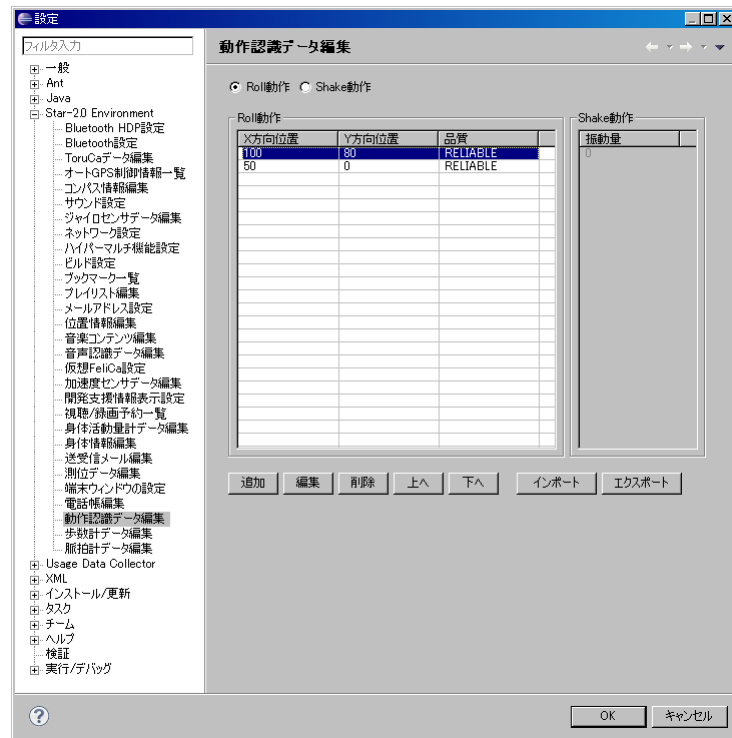


図 19-80 動作認識データ編集画面

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 19-81）。



図 19-81 動作認識データ設定画面

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、「7.11. 動作認識データ編集」を参照してください。

19.4.24. 音声認識データ編集

音声認識機能の認識データを編集します。

エミュレータではバックエンドサーバに対し実際に DDP 通信は起こりません。本設定をおこなうことにより、指定したとおりに DDP 通信のレスポンスをエミュレートできます。また、特微量データは一定間隔(デフォルトは 240msec)毎にダミー値を返します。(ダミー値は音声取得情報設定ファイルの値を直接変更することにより調節可能です)

音声認識データを編集するには：

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、「Star-2.0 Environment」→「音声認識データ編集」を選択します（図 19-82）。

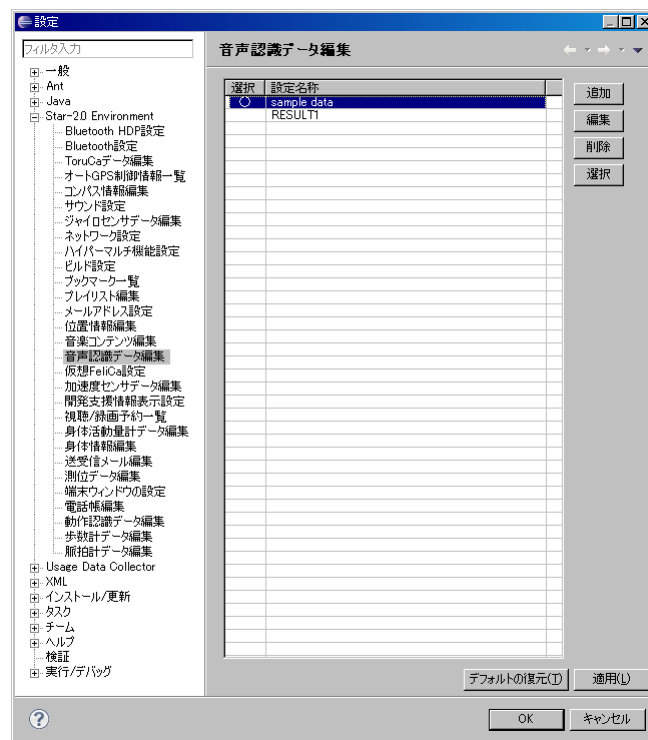


図 19-82 音声認識データ編集画面

3. API で使う認識結果データを選択する場合は、認識結果データを一覧から選択して「選択」ボタンを押します。
4. 認識結果データを削除したい場合は、一覧から削除した認識結果データを選択して「削除」ボタンを押します。
5. 音声認識データを追加したい場合は「追加」、音声認識データを編集したい場合は一覧から編集したい音声認識データを選択して「編集」ボタンを押すと、認識結果情報詳細画面が表示されます（図 19-83）。

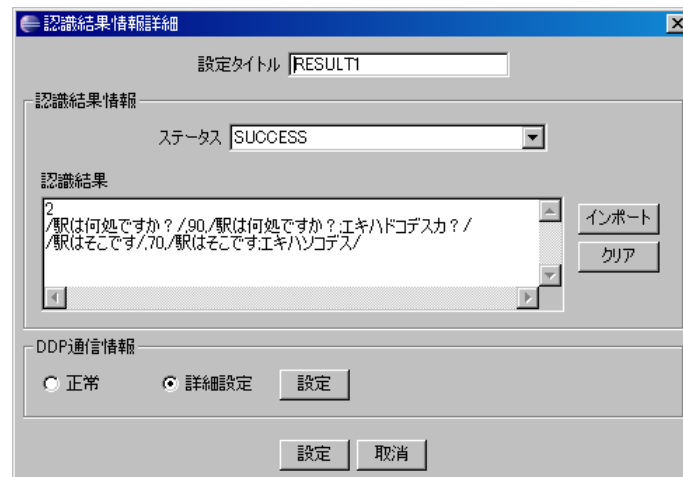


図 19-83 認識結果情報詳細画面

6. 各設定項目を設定します。
7. 「設定」を押すと、音声認識データ編集画面に追加されます。
8. 「インポート」ボタンを押すと、認識結果をファイルから読み込みます。
9. 「クリア」ボタンを押すと、認識結果の入力をクリアします。
10. DDP通信情報の「設定」を押すと、DDP通信を実行した結果の詳細設定画面を表示します（図 19-84）。
もし、DDP通信の振る舞いを設定したい場合は、各種コマンド発行時のエラー情報（サーバレスポンスコード）を指定してください。

DDP通信情報編集

ネゴシエーション
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

送信
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG
送信回数 ☒ 指定なし ☐ 指定あり 1 回

結果取得
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

キャンセル
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

終了
結果 ☐ 正常 ☒ 異常
エラー情報 400 NG

設定 取消

図 19-84 DDP 通信情報編集画面

11. 各設定項目を設定して、「設定」ボタンを押します。

各設定項目の詳細については、「7.12. 音声認識データ編集」を参照してください。

19.4.25. 歩数計データ編集

歩数計データを編集するには：

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、「Star-2.0 Environment」→「歩数計データ編集」を選択します（図 19-85）。



図 19-85 歩数計データ一覧画面

3. 歩数計データを追加したい場合は「追加」、歩数計データを編集したい場合は一覧から編集したい歩数計データを選択して、「編集」ボタンを押します（図 19-86）。

歩数計データ一覧

日付 2010 年 08 月 01 日

周期 3600 秒

時刻	歩数	歩行距離	有酸素運動の歩数	有酸素運動の時間	燃焼脂肪量	消費カロリー
00:00	0	0	0	0	0	0
01:00	0	0	0	0	0	0
02:00	0	0	0	0	0	0
03:00	0	0	0	0	0	0
04:00	0	0	0	0	0	0
05:00	0	0	0	0	0	0
06:00	0	0	0	0	0	0
07:00	0	0	0	0	0	0
08:00	300	175	60	2	400	190
09:00	600	350	120	5	800	380
10:00	600	350	120	10	800	380
11:00	4500	2625	0	0	6000	2850
12:00	0	0	0	0	0	0
13:00	0	0	0	0	0	0
14:00	0	0	0	0	0	0
15:00	0	0	0	0	0	0
16:00	0	0	0	0	0	0
17:00	0	0	0	0	0	0
18:00	0	0	0	0	0	0
19:00	0	0	0	0	0	0
20:00	0	0	0	0	0	0
21:00	0	0	0	0	0	0

図 19-86 歩数計データ一覧画面

4. 歩数計データを追加したい場合は、まず周期表示の右側の「設定」ボタンを押して、日付と周期を設定します（図 19-87）。

歩数計基本情報設定

日付 2010 年 8 月 1 日

周期 3600 秒

図 19-87 歩数計基本情報設定画面

5. 各設定項目を設定して「設定」を押すと、歩数計データ一覧画面に戻り、歩数計データの設定が可能になります。
6. 一覧から編集したい時刻のデータを選択して「編集」ボタンを押します（図 19-88）。

歩数計データ設定

歩数

歩行距離 m

有酸素運動の歩数

有酸素運動の時間 分

燃焼脂肪量 g

消費カロリー Kcal

図 19-88 歩数計データ設定画面

7. 歩数計データを設定して「設定」を押すと、歩数計データ一覧画面に戻ります。
8. 編集が完了したら、歩数計データ設定画面で「設定」を押すと、歩数計データ編集画面に戻ります。
9. 歩数計データを削除したい場合は、一覧から削除したい歩数計データを選択して「削除」ボタンを押します。

各設定項目の詳細については、「7.13. 歩数計データ編集」を参照してください。

19.4.26. 身体活動量計データ編集

身体活動量計データを編集するには:

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、「Star-2.0 Environment」→「身体活動量計データ編集」を選択します（図 19-89）。

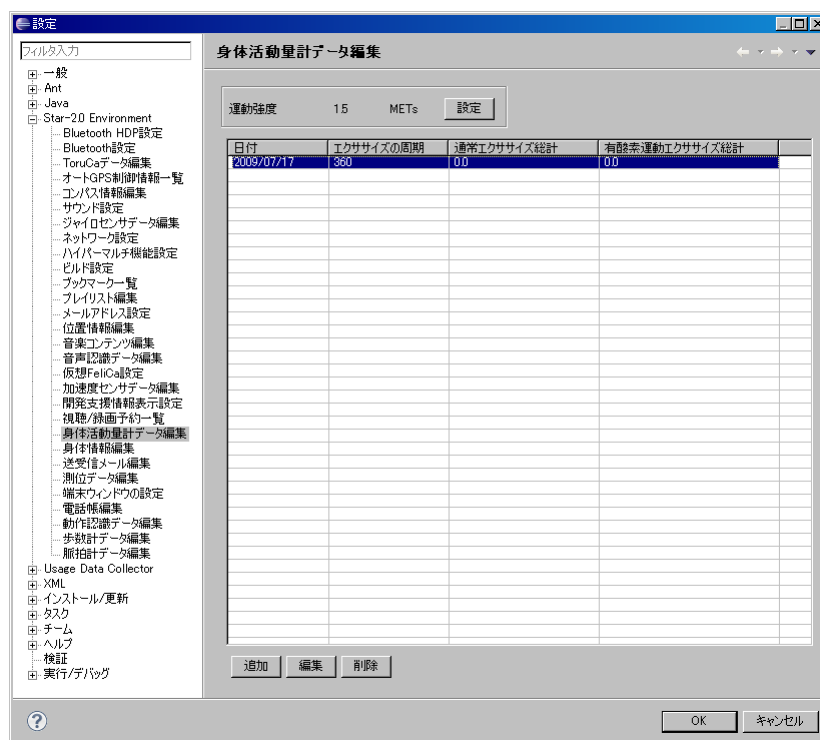


図 19-89 身体活動量計データ一覧画面

3. 身体活動量計データを追加したい場合は「追加」、身体活動量計データを編集したい場合は一覧から編集したい身体活動量計データを選択して、「編集」ボタンを押します（図 19-90）。

時刻	通常エクササイズ ^g	有酸素運動エクササイズ ^g
00:00	0	0
01:00	0	0
02:00	0	0
03:00	0	0
04:00	0	0
05:00	0	0
06:00	0	0
07:00	0	0
08:00	0	0
09:00	0	0
10:00	0	0
11:00	0	0
12:00	0	0
13:00	0	0
14:00	0	0
15:00	0	0
16:00	0	0
17:00	0	0
18:00	0	0
19:00	0	0
20:00	0	0
21:00	0	0
22:00	0	0

図 19-90 身体活動量計データ設定画面

4. 身体活動量計データを追加したい場合は、まず周期表示の右側の「設定」ボタンを押して、日付と周期を設定します（図 19-91）。

図 19-91 身体活動量計基本情報設定画面

5. 各設定項目を設定して「設定」を押すと、身体活動量計データ設定画面に戻り、身体活動量計データの設定が可能になります。
6. 一覧から編集したい時刻のデータを選択して「編集」ボタンを押します（図 19-92）。

図 19-92 エクササイズ設定画面

7. エクササイズを設定して「設定」を押すと、身体活動量計データ設定画面に戻ります。

8. 編集が完了したら、身体活動量計データ設定画面で「設定」を押すと、身体活動量計データ編集画面に戻ります。
9. 運動強度を変更したい場合は、身体活動量計データ編集画面で「設定」ボタンを押します（図 19-93）。



図 19-93 運動強度設定画面

10. 運動強度を設定して「設定」を押すと、身体活動量計データ編集画面に戻ります。
11. 身体活動量計データを削除したい場合は、一覧から削除したい身体活動量計データを選択して「削除」ボタンを押します。

各設定項目の詳細については、「7.14 身体活動量計データ編集」を参照してください。

19.4.27. オート GPS 制御情報一覧

オート GPS 制御情報を参照するには:

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、「Star-2.0 Environment」→「オートGPS制御情報一覧」を選択します（図 19-94）。

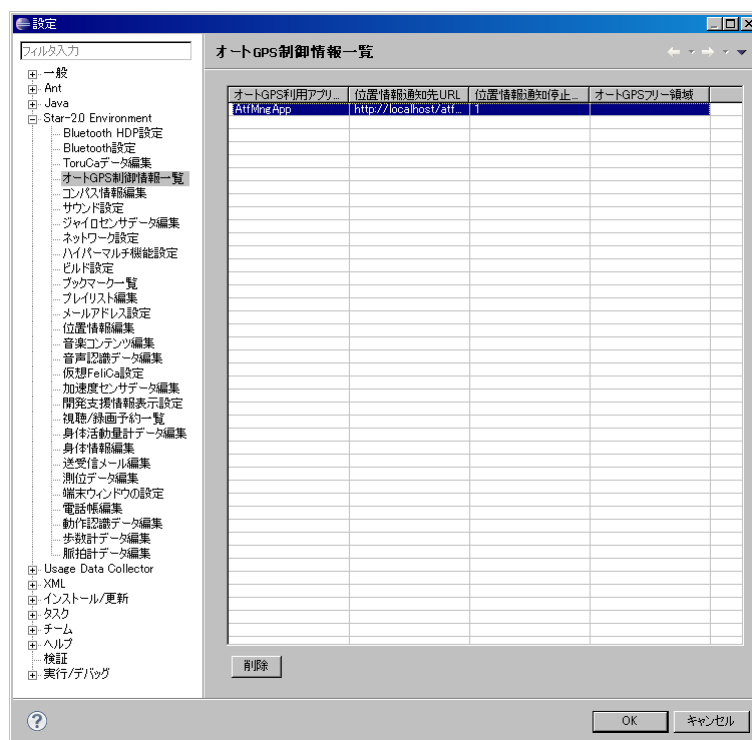


図 19-94 オート GPS 制御情報一覧画面

3. オート GPS 制御情報を削除したい場合は、一覧からオート GPS 制御情報データを選択し、「削除」を押します。

19.4.28. ジャイロセンサデータ編集

ジャイロセンサデータを編集するには：

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、「Star-2.0 Environment」→「ジャイロセンサデータ編集」を選択します（図 19-95）。

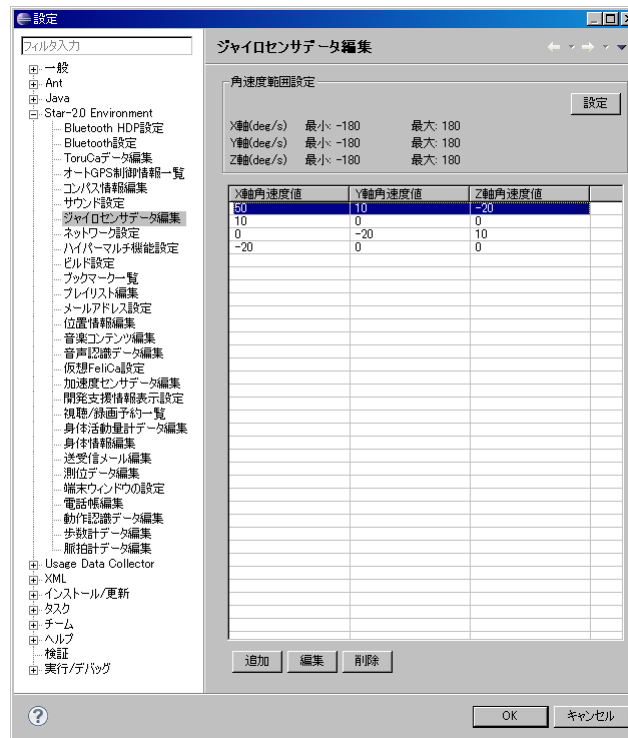


図 19-95 ジャイロセンサデータ一覧画面

3. ジャイロセンサデータを追加したい場合は「追加」、ジャイロセンサデータを編集したい場合は一覧から編集したいジャイロセンサデータを選択して、「編集」ボタンを押します（図 19-96）。

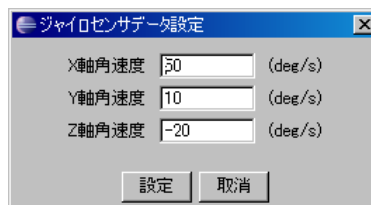


図 19-96 ジャイロセンサデータ設定画面

4. X 軸、Y 軸、Z 軸の角速度を入力後、「設定」を押すと、ジャイロセンサデータ編集画面に戻ります。
5. 角速度範囲設定を変更したい場合は、ジャイロセンサデータ編集画面で「設定」ボタンを押します（図 19-97）。

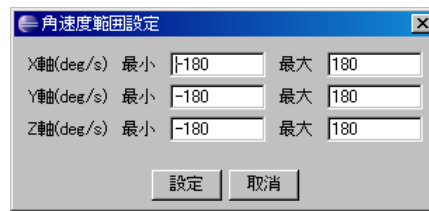


図 19-97 角速度範囲設定画面

6. X 軸、Y 軸、Z 軸の角速度範囲を設定して「設定」を押すと、ジャイロセンサデータ編集画面に戻ります。
7. ジャイロセンサデータを削除したい場合は、一覧から削除したいジャイロセンサデータを選択して「削除」ボタンを押します。

各設定項目の詳細については、「7.16 ジャイロセンサデータ編集」を参照してください。

19.4.29. 身体情報編集

身体情報データを編集するには：

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、「Star-2.0 Environment」→「身体情報編集」を選択します（図 19-98）。



図 19-98 身体情報編集画面

3. 身体情報データを編集したい場合は「身長」、「体重」の各項目に値を入力します。

各設定項目の詳細については、「7.17 身体情報編集」を参照してください。

19.4.30. 脈拍計データ編集

脈拍計データを編集するには：

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、「Star-2.0 Environment」→「脈拍計データ編集」を選択します（図 19-99）。

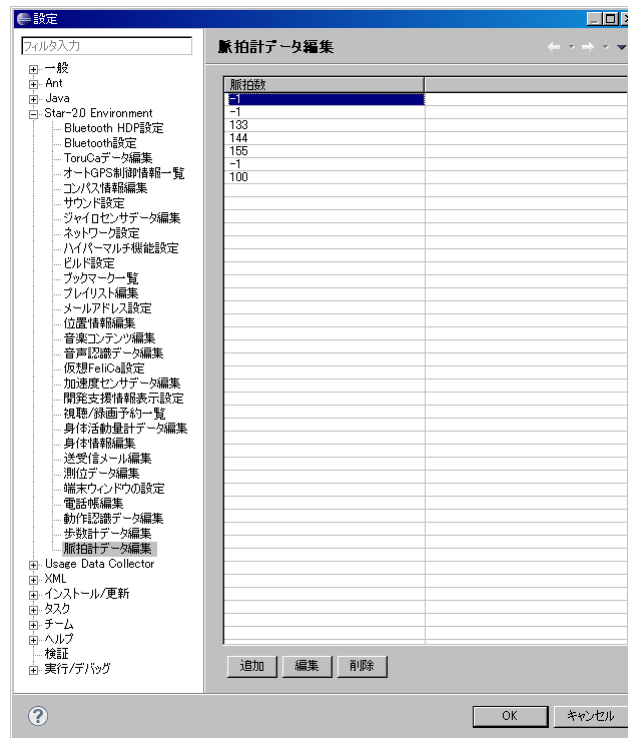


図 19-99 脈拍計データ編集画面

3. 脈拍計データを追加したい場合は「追加」、脈拍計データを編集したい場合は一覧から編集したい脈拍計データを選択して、「編集」ボタンを押します（図 19-100）。



図 19-100 脈拍計データ設定画面

4. 脈拍数を入力後、「設定」を押すと、脈拍計データ編集画面に戻ります。
5. 脈拍計データを削除したい場合は、一覧から削除したい脈拍計データを選択して「削除」ボタンを押します

各設定項目の詳細については、「7.18 脈拍計データ編集」を参照してください。

19.4.31. 音楽コンテンツ編集

音楽コンテンツを編集するには：

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、「Star-2.0 Environment」→「音楽コンテンツ編集」を選択します（図 19-101）。

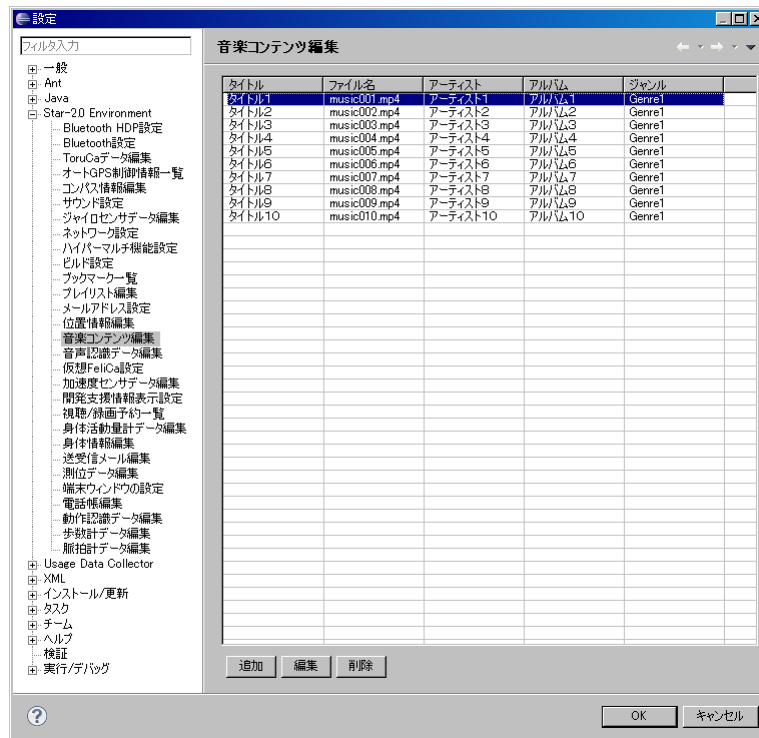


図 19-101 音楽コンテンツ編集画面

3. 音楽コンテンツを追加したい場合は「追加」、音楽コンテンツを編集したい場合は一覧から編集したい音楽コンテンツを選択して、「編集」ボタンを押します（図 19-102）。

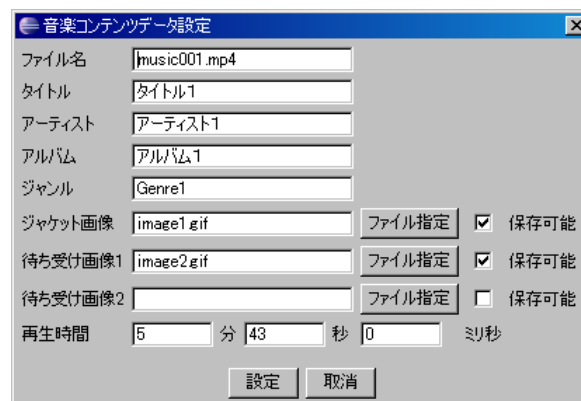


図 19-102 音楽コンテンツデータ設定画面

4. 音楽コンテンツデータを入力後、「設定」を押すと、音楽コンテンツ編集画面に戻ります。
5. 音楽コンテンツを削除したい場合は、一覧から削除したい音楽コンテンツを選択して「削除」ボタンを押します。

各設定項目の詳細については、「7.19 音楽コンテンツ編集」を参照してください。

19.4.32. プレイリスト編集

プレイリストを編集するには：

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、「Star-2.0 Environment」→「プレイリスト編集」を選択します（図 19-103）。

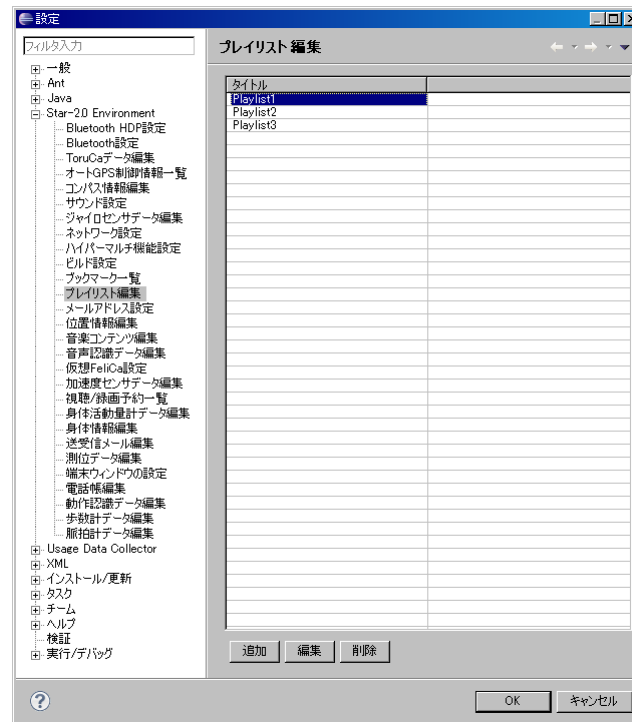


図 19-103 プレイリスト編集画面

3. プレイリストを追加したい場合は「追加」、プレイリストを編集したい場合は一覧から編集したいプレイリストを選択して、「編集」ボタンを押します（図 19-104）。

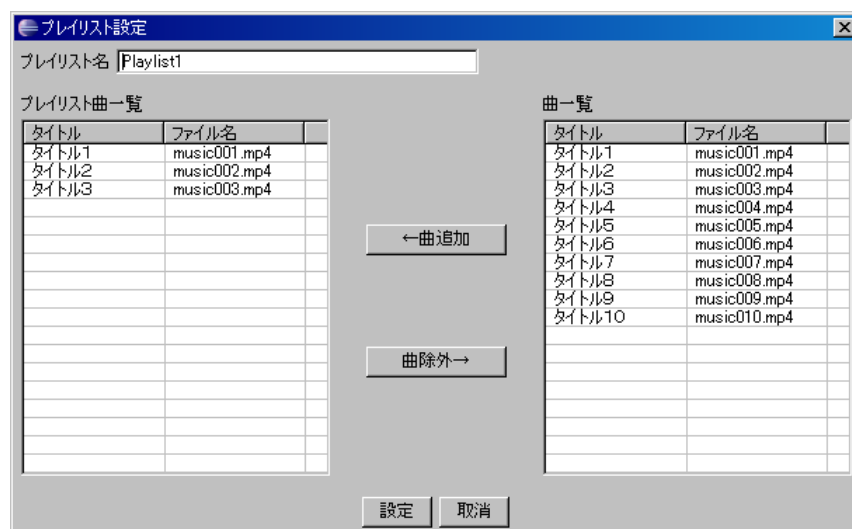


図 19-104 音楽コンテンツデータ設定画面

4. プレイリストデータを入力後、「設定」を押すと、プレイリスト編集画面に戻ります。
5. プレイリストを削除したい場合は、一覧から削除したいプレイリストを選択して「削除」ボタンを押します。

各設定項目の詳細については、「7.20 プレイリスト編集」を参照してください。

19.5. 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 データファイルの内容を表示・編集するためのツールです。

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事ができます。

階層表示及びバイナリ表示の2つの表示モードが利用可能ですが、階層表示については仮想 FeliCa 動作設定が DoJa-5.0 用に設定されている場合のみ利用可能となります。

19.5.1. 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されます：

<プロジェクトコンテンツフォルダ>¥<プロジェクト名>.fc

19.5.2. 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「プロジェクト」→「Star-2.0」→「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 19-105）。

インストール直後のデータファイルにはデータ領域を除き適切な値が設定されています。仮想FeliCaデータファイルの構造については、「15. 29. 仮想FeliCaデータファイル」を参照してください。

「表示切替」ボタンを押す事で、階層表示～バイナリ表示の切替が可能です。

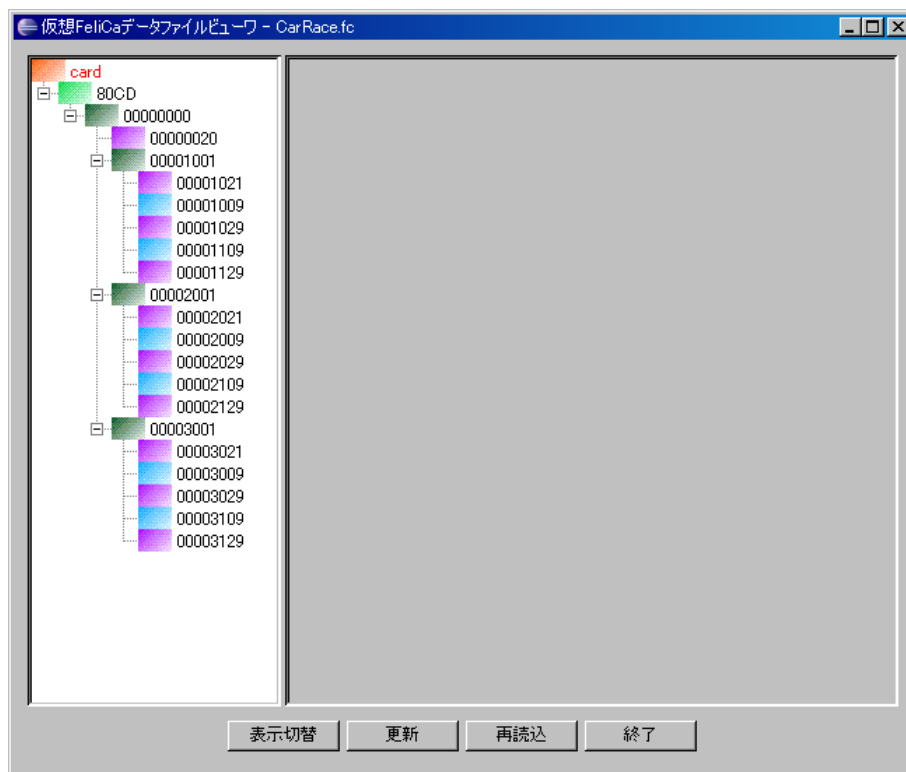


図 19-105 仮想 FeliCa データファイルビューワ

19.5.3. 階層表示

階層表示では、仮想FeliCaデータのブロックの関係をツリー構造で表示します（図 19-106）。

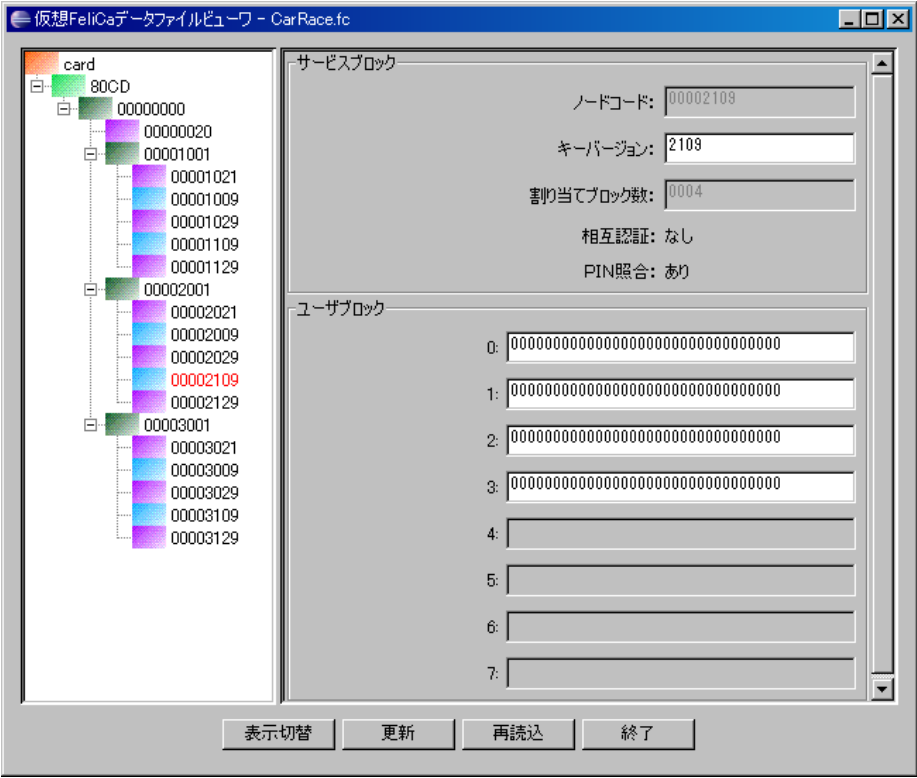


図 19-106 仮想 FeliCa データファイルビューワ（階層表示）

(1) ノードの表示
各ノードを示すアイコン及び表示色は表 19-1 のようになります。

表 19-1 階層表示におけるノード表示

アイコン	表示色	ノード
	橙	カード
	黄緑	システム
	緑	エリア
	水色	サービス
	紫	PIN サービス

(2) ノードのブロック情報表示
ノードコードツリーで選択されているノードに関連するブロック情報を表示します。
選択中のノードによって、表示される情報は表 19-2 のようになります。

表 19-2 選択ノードのブロック情報表示項目

選択中のノード	表示項目	編集
カード	—	—
システム	システムコード キーバージョン	× ○
エリア	ノードコード キーバージョン エンドサービスコード 子エリア（の作成可・不可） PIN 照合（の有無）	× ○ × × ×
サービス	ノードコード キーバージョン 割当ブロック数 相互認証（の有無） PIN 照合（の有無） ユーザブロック	× ○ × × × ○
PIN サービス	ノードコード キーバージョン PIN 照合試行回数 照合試行上限回数 PIN 属性（B8） PIN 属性（Bc）	× ○ ○ × × × ×

(3) ノード情報の編集

ノード情報を編集するには：

1. ノードコードツリーから任意のノードを選択すると、ビューワ右部に選択ノードのブロック情報が表示されます。
2. ブロック情報の任意の内容を変更し、ノードコードツリーの他のノードを選択するか、「更新」ボタンを押すと確定されます。

- ※ テキストフィールドには値は全て 16 進数で入力してください。
- ※ 各項目を変更し、ノードコードツリーの他のノードを選択して編集を確定した時点では、仮想 FeliCa データファイルには反映されていません。仮想 FeliCa データファイルへの反映は「更新」ボタンを押してください。
- ※ 階層表示では、キーバージョン、ユーザブロック、PIN を編集することができます（表 19-2）。
- ※ 「再読込」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。
- ※ 「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。

19.5.4. バイナリ表示

バイナリ表示では、仮想FeliCaデータのブロックをバイナリ形式で表示します（図 19-107）。

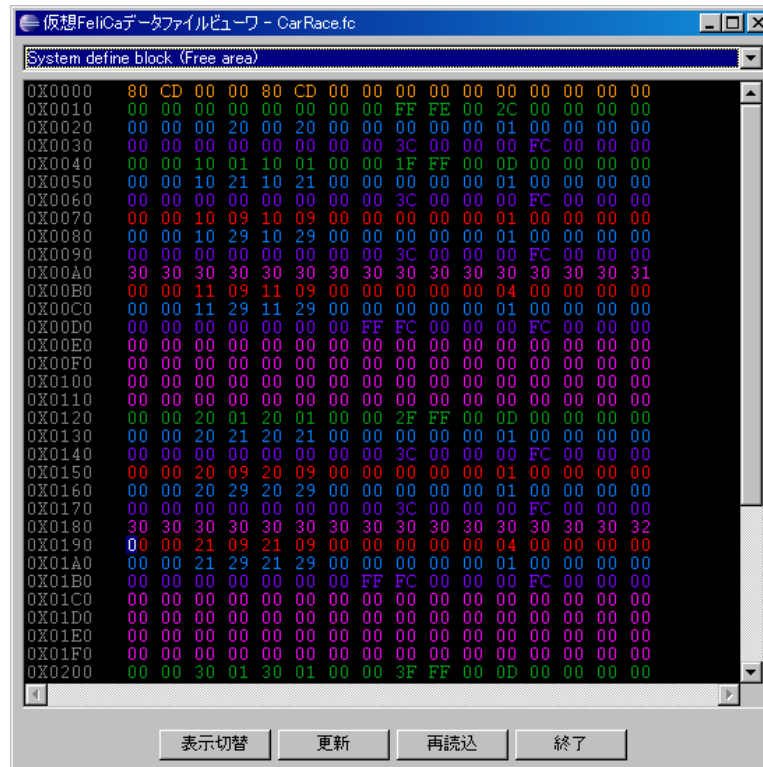


図 19-107 仮想 FeliCa データファイルビューワ（バイナリ表示）

(1) ブロックの表示

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

(2) 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

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

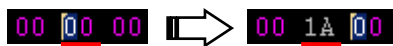
仮想 FeliCa データファイルビューワで編集を行う場合のエンディアン表記については、全てビッグエンディアンになっています。

(3) ブロックの編集

ブロックを編集するには：

1. 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。

2. 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。

3. 「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。
4. 「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。
5. 「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。

19.5.5. フリー領域を扱うアプリの利用

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、ADFのFeliCaCPIDと同じ値でなければなりません（「4.8. 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（「4.8. 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（「19.3.5.2. FeliCaフリー領域の関連付け」参考）。

階層表示で手動設定するには：

1. ノードコードツリーから サービスノード 0000*009 を選択します。
*は 1～3 の任意のノードコード
2. CPID サービスのブロックの情報が表示されるので、ユーザブロックを ADF に設定した FeliCaCPID と同じ値に変更します。
3. ノードコードツリーから PIN サービスノード 0000*129 を選択します。
*は 1～3 の任意のノードコードで、手順 1 と同じ値を指定します
4. PIN サービスのブロックの情報が表示されるので、PIN を ADF に設定した FeliCaPIN の値に変更します。例えば、ADF に設定した FeliCaPIN の値が 12345678 の場合は、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。

バイナリ表示で手動設定するには：

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、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。

19.6. スクラッチパッド

19.6.1. 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。

19.6.2. 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表示するには：

1. メニューから「プロジェクト」→「Star-2.0」→「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 19-108）。

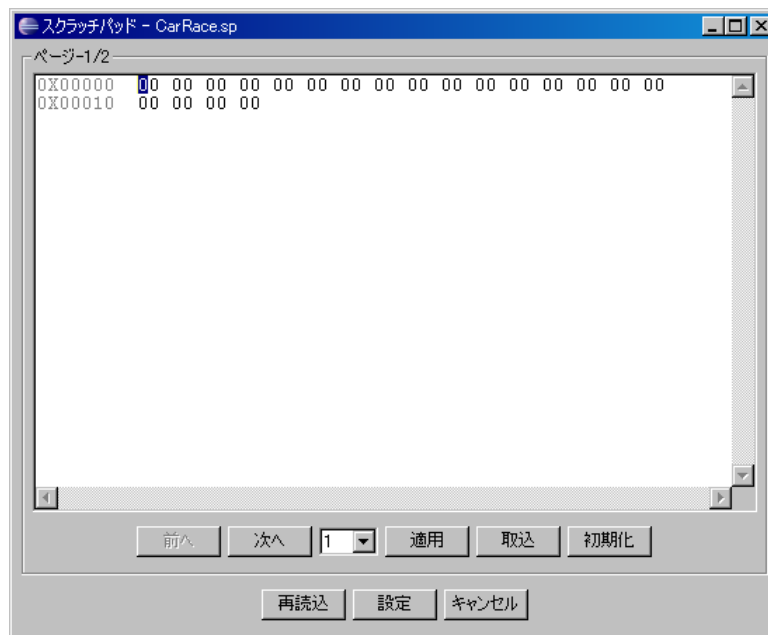
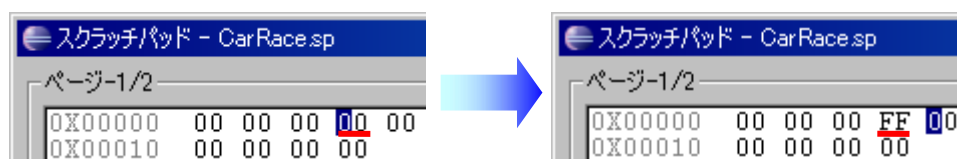


図 19-108 スクラッチパッド管理ツール画面

2. 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。
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。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 つ前のスク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 つ先のスクラッチパッドの内容が表示されます。
6. プルダウンからスクラッチパッドの連番を選択すると、そのスクラッチパッドの内容が表示されます（図 19-109）。



図 19-109 スクラッチパッドの選択

7. 「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。
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。
まず、取り込みたい位置を選択します（図 19-110）。

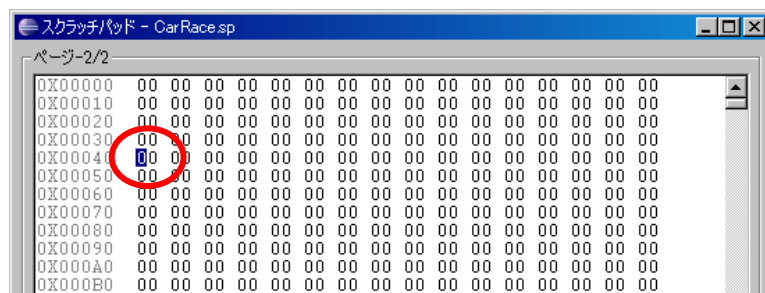


図 19-110 スクラッチパッド 置換開始位置選択

「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 19-111）。

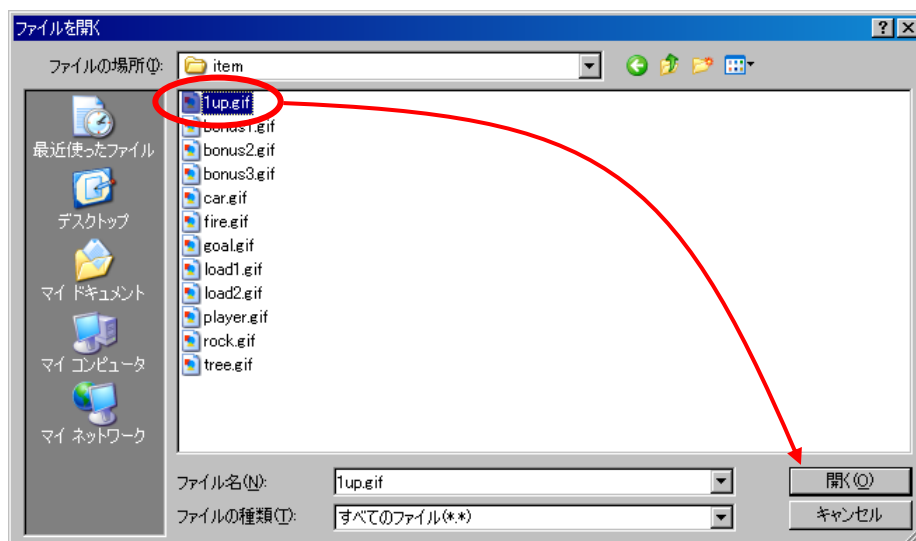


図 19-111 置換ファイル選択ダイアログ

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 19-112）。
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。

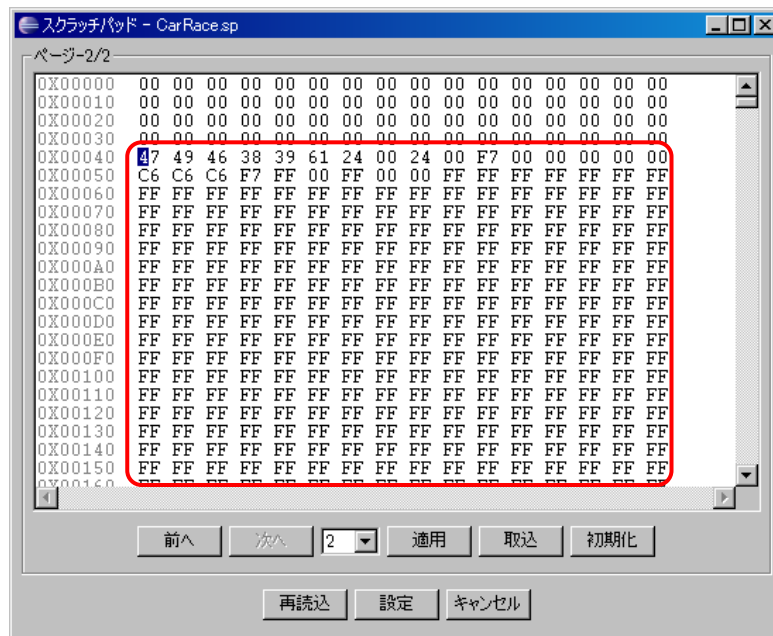


図 19-112 置換後のスクラッチパッド

9. 「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。
10. 「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。
11. 「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。
12. 「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。

19.6.3. 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」→「Star-2.0」→「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。

19.6.4. 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ファイル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（「4.8. 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。

19.7. ADF チェック

ADF チェックツールは、HTTP サーバ上に配置された i アプリの ADF (.jam ファイル) 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、ADF 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場合があります。

本ツールでは、こうした現象を解消するため、HTTP サーバ上に設置された i アプリの ADF を読み取ってチェックする事ができます。

19.7.1. ADF チェックツールの起動

ADF チェックツールを起動するには：

1. メニューから「プロジェクト」→「Star-2.0」→「ADFチェック...」を選択します。「ADFチェックツール」画面が表示されます（図 19-113）。

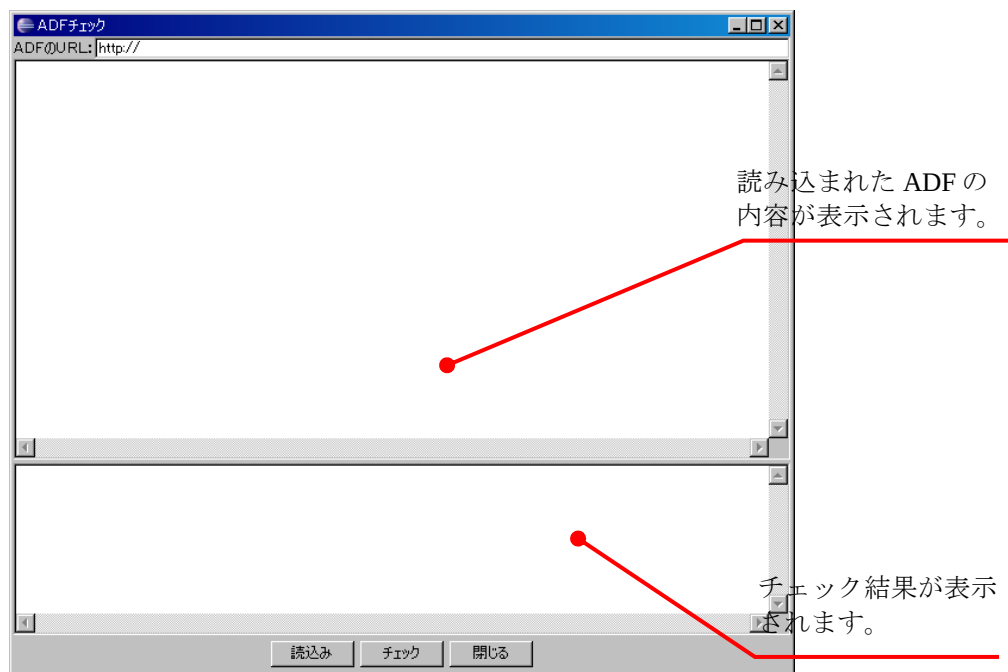


図 19-113 ADF チェックツール

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:/iDKStar2.0/apps/CarRace/bin/CarRace.jam)
3. 「読み込み」ボタンを押すと ADF を HTTP 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 19-114）。

「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。

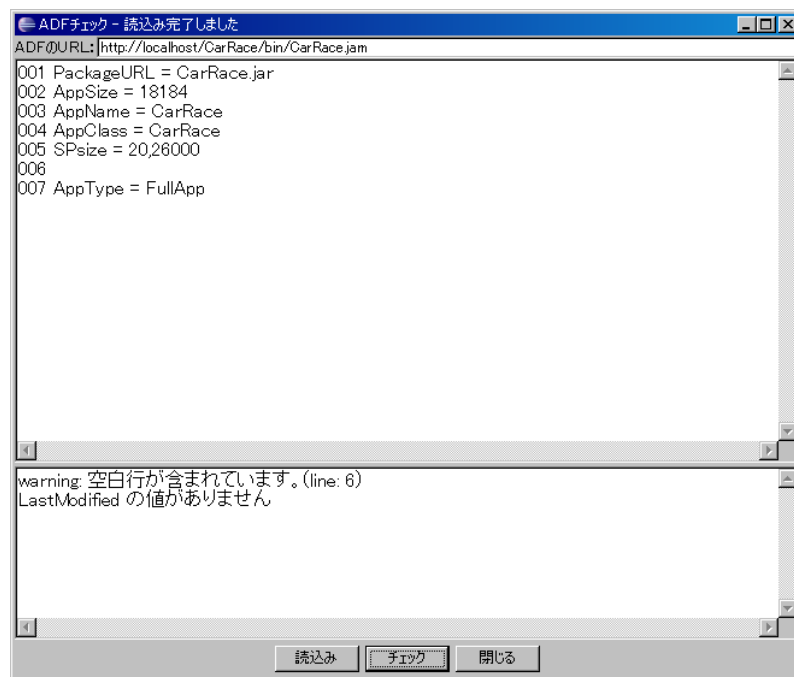


図 19-114 ADF のチェック実行

- ※ 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 ファイル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 が正しいかどうかも確認してみてください。

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¥device1¥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
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` = [3D 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 ～ 8

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 3D Track 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。

2. OpenGL/ES 機能の設定

OpenGL/ES 機能のプロパティを指定できます。

`egl_depth_size` = [Z バッファの深度]

指定値：0 ～ 32

未指定：32

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。	13,30
エミュレーション端末	エミュレータの携帯電話画面。	30,266,264,251
i アプリ	i モード携帯電話で動作する Java アプリケーション (ソフトウェア)。	34
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、API 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。	34,91
i アプリ DX	「トラステッド i アプリ」に同じ。	-
DoJa	i アプリの Java 仕様 (API) 。 i アプリでは、CLDC1.1 と DoJa プロファイルの API が扱えます。	13
Star	i アプリの次期 Java 仕様 (API) 。 i アプリでは、CLDC1.1 と Star プロファイルの API が扱えます。	13
CLDC	Connected Limited Device Configuration DoJa および Star が動作する組み込み機器用の Java 環境 (J2ME)。	13
J2SE JDK	Java(TM) 2 SDK, Standard Edition Oracle 社の Java2 の開発環境で、Oracle 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。	13
DirectX	Microsoft 社の Windows 用拡張 API で、Microsoft 社から取得してインストールできます。 → Windows Update	13
JAM	Java Application Manager i アプリのダウンロードや起動を制御する機能。	63
ADF	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。	63
サスペンド	i アプリが通話中などで中断している状態。	266
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。	266
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。	35,63,285,347
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 および Star 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。	156
コード認識	QR コードやバーコードの認識機能。	104,331

用語	説明	関連ページ
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。	107,109,111,191, 177,336
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。	194,336
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)	63,85,86,173, 290,346
FeliCa アドホック	FeliCa を用いた無線通信機能。 ※ DoJa-5.0 から導入	113,173,338
トルカ	FeliCa を用いた電子カード機能。 ※ DoJa-4.1 から導入	116,201,334
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。	164
OBEX	OBject EXchange 無線通信のプロトコル。	153,368
IrDA	Infrared Data Association 赤外線通信の規格。	63,368
Bluetooth	携帯電話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。	166,338
DTV	Digital Television デジタル変調・圧縮の技術を利用したテレビ放送。	122,342
BML	Broadcast Markup Language データ放送向けのページ記述言語。	120,342
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。	120,342
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。	209
加速度センサ	携帯電話の移動速度の変化を計測する回路素子。	206
ジャイロセンサ	携帯電話の角速度の変化を計測する回路素子。	225
Eclipse	幅広く使われているオープンソースの統合開発環境。 (無償)	14,370
音声認識	入力された音声を文字列へと変換する機能。	212,348,438
特徴量データ	音声波形から抽出した音声の特徴データ。	348
DDP 通信	DoCoMo DSR Protocol	212,438
DSR	Distributed Speech Recognition (分散型音声認識)	-
Widget View	Star プラットフォームで動作するミニアプリの実行環境。フルアプリ同様に VM として動作します。	35
ミニアプリ	Widget View 上で動作する i アプリ。フルアプリより機能が劣りますが、最大 8 件まで同時起動することが可能です。	35
フルアプリ	Star 向けの i アプリ。	35
招集機能	SMS を利用して、遠隔のサーバや他の i モード携帯電話からの要求でアプリを起動する機能。	57,91,126