

先行 AudioPresenter : ポート指定なし 後行 AudioPresenter : ポート指定なし の場合



- ① : AP-A はポート指定なしで取得されているため、利用できるリソースは通常の AudioPresenter の通りである。AP-A で再生される MF*i* は音源チップのリソースをフルに利用して再生される。
- ② : ここでポート指定なしで取得された AP-B の再生が指示される。この時点において、2 つの MF*i* で使用するリソースの合計が音源チップの範囲内である場合は複数同時再生を行う必要が発生するが、N900i (S) ではこれをサポートしない。N900i (S) では、AP-B の再生指示を受け付けると AP-A の再生を停止する。これは完全な停止であり、AP-B の再生が完了しても自動的に継続または再開がされることはない (アプリ側で明示的に AP-A に再生指示を出す必要がある)。
- ③ : AP-B の再生が完了する。AP-A の方は停止状態、AP-B の方は完了状態となる。

先行 AudioPresenter : ポート指定あり 後行 AudioPresenter : ポート指定なし の場合



- ① : AP-A はポート指定つきで取得されているため、利用できるリソースは通常の AudioPresenter の半分である。AP-A で再生される MF*i* はこの制限下で再生される。
- ② : ここでポート指定なしで取得された AP-B の再生が指示される。DoJa-3. x プロファイルでは、ポート指定つきの AudioPresenter とポート指定なしの AudioPresenter を同時に再生することはできない。ポート指定つきの AP-A を再生中にポート指定なしの AP-B を再生させようとする、API リファレンスにおける規定に基づき AP-A が停止して AP-B の再生が開始される。AP-A の停止は完全な停止であり、AP-B の再生が完了しても自動的に継続または再開がされることはない (アプリ側で明示的に AP-A に再生指示を出す必要がある)。AP-B はポート指定なしで取得されているため通常の AudioPresenter と同等であり、AP-B で再生される MF*i* は音源チップのリソースをフルに利用して再生される。
- ③ : AP-B の再生が完了する。AP-A の方は停止状態、AP-B の方は完了状態となる。

先行 AudioPresenter : ポート指定なし 後行 AudioPresenter : ポート指定あり の場合



- ① : AP-A はポート指定なしで取得されているため、利用できるリソースは通常の AudioPresenter の通りである。AP-A で再生される MF*i* は音源チップのリソースをフルに利用して再生される。
- ② : ここでポート指定つきで取得された AP-B の再生が指示される。DoJa-3. x プロファイルでは、ポート指定つきの AudioPresenter とポート指定なしの AudioPresenter を同時に再生することはできない。ポート指定なしの AP-A を再生中にポート指定つきの AP-B を再生させようとする、API リファレンスにおける規定に基づき AP-A が停止して AP-B の再生が開始される。AP-A の停止は完全な停止であり、AP-B の再生が完了しても自動的に継続または再開がされることはない (アプリ側で明示的に AP-A に再生指示を出す必要がある)。AP-B はポート指定つきで取得されているため利用できるリソースは通常の AudioPresenter の半分であり、AP-B で再生される MF*i* はこの制限下で再生される。
- ③ : AP-B の再生が完了する。AP-A の方は停止状態、AP-B の方は完了状態となる。

2. N900i カメラ機能呼び出しにおけるフレーム設定機能の制限

com.nttdocomo.device.Camera クラスでは、setFrameImage() メソッドを使用して i アプリが任意のフレーム画像を設定した状態でカメラ機能呼び出すことができます。これについて、N900i では i アプリからのカメラ起動においてフレーム画像の設定をサポートしません。N900i で Camera.setFrameImage() メソッドを呼び出すと UnsupportedOperationException が発生します。N900i 向けの i アプリでは、フレーム設定機能を使用しないようにしてください。なお、この制限は N900iS では改善されます。

3. N900i 待ち受けアプリケーションにおけるバイブレータ制御

N900i では、非活性化状態の待ち受けアプリケーションからバイブレータ制御を行うことはできません。N900i では、非活性化状態の待ち受けアプリケーションから PhoneSystem.DEV_VIBLATOR を指定して PhoneSystem.setAttribute() を呼び出しても無視されます。なお、この制限は N900iS では改善されます。

4. N900i (S) ヒープ上のフラグメントによるメモリ不足の発生

N900i (S) に搭載されている KVM では、Java ヒープ上に発生したフラグメント（未使用領域の断片）をシステム側で回収、結合するヒープコンパクション機能がサポートされていません。このため、HTTP 通信・ScratchPad などの入出力処理やイメージ処理など、一時的なバッファを多用する処理を繰り返し実行していると、ヒープ全面にフラグメントが発生してしまい以降の処理で以下のような例外やエラーが発生しやすくなります。

- ・ OutOfMemoryError
- ・ ConnectionException (NO_RESOURCE)
- ・ UIException (NO_RESOURCES)

(フラグメントの存在によって連続バッファ領域の確保が阻害されることが原因となるため、多くの場合 Runtime.freeMemory() の返すヒープ空き総量は十分大きな値を示します。)

過去の多くの機種でも Java ヒープのコンパクション機能はサポートされていませんが、900i シリーズの i アプリ実行環境では通信サイズや ScratchPad の容量拡大に伴いシステムが大きなバッファを要求する契機が多くなったこともあり、N900i (S) では上記例外やエラーが発生しやすくなる傾向があります。

本問題については、

- ・ メモリ使用量の多い処理の前後で、GC をよりこまめに起動する。
- ・ 使用頻度の高いイメージについては、使用の都度ロード、破棄するのではなく i アプリ起動時にロードしておく。また使用頻度の低いイメージについては常時ロードしておくことは避け、ロードされたイメージがメモリを過度に圧迫することのないようにする。

といった対策によりフラグメントの増加度合いをできる限り小さく抑えることが一般的な対処方法となります。また、ScratchPad 入出力で上記のエラーが発生するケースでは、ScratchPad の URL に length オプションを付加することで、ScratchPad アクセス時に内部で使用する入出力バッファのサイズが小さくなり問題を回避できる場合があります。