

テキスト入出力と時間計測

1. 概要

アプリケーションの作成において、テキストデータに関するファイル入出力やコンソールへの出力はよく使われる機能です。このような機能はアプリケーションの動作を確認するためにも重要な役割を果たします。また、動作を確認するという点では、各処理にかかる時間を計測する機能も重要です。ここでは、これらの機能を実現するコンポーネントについて説明します。

この文書で説明するコンポーネントのリストとそれらの機能は次の通りです。

- テキストファイル入力： ファイルからテキストデータを読み込む機能
- テキストファイル出力： ファイルにテキストデータを書き出す機能
- 標準入力： コンソールからテキストデータを読み込む機能
- 標準出力： コンソールにテキストデータを書き出す機能
- 標準エラー出力： 標準出力と同様の機能（エラー出力であることを明示する場合）
- 時間計測： アプリケーション内部の各処理にかかる時間を計測する機能

また、これらをビルダーで追加する際のメニュー内の位置は次の通りです。

- テキストファイル入力： [コンポーネント追加]-[入出力]-[ファイル]-[テキストファイル入力]
- テキストファイル出力： [コンポーネント追加]-[入出力]-[ファイル]-[テキストファイル出力]
- 標準入力： [コンポーネント追加]-[入出力]-[システム]-[標準入力]
- 標準出力： [コンポーネント追加]-[入出力]-[システム]-[標準出力]
- 標準エラー出力： [コンポーネント追加]-[入出力]-[システム]-[標準エラー出力]
- 時間計測： [コンポーネント追加]-[処理部品]-[システム]-[時間計測]

2. 用途

この文書で説明するコンポーネント群の用途として以下のようなものが挙げられます。()内は主に関係するコンポーネントを示しています。

- テキストエディタの簡易版を作成する (テキストファイル入力・テキストファイル出力)
- アプリケーションの実行状況をコンソールやログファイルに出力する (テキストファイル出力・標準出力・標準エラー出力)
- コンソールアプリケーションを作成する (標準入力・標準出力・標準エラー出力)
- アプリケーションの動作を解析する (時間計測・標準出力)

3. ここで使用されるイベントとメソッド

各コンポーネントに関して、この文書内で使用されるイベントとメソッドの一覧を記します。ここに記す以外にもイベントやメソッドがありますが、それらの情報が必要な場合は各コンポーネントのリファレンスや Javadoc ドキュメントを参照してください。

- テキストファイル入力
 - データ生成イベント

イベント発生条件	内包データ	イベント番号
ファイルから全体のテキストデータ読み込み	ファイル全体のテキストデータ (String)	0
ファイルから行単位のテキストデータ読み込み	一行分のテキストデータ (String)	1
ファイルから単語単位のテキストデータ読み込み	一単語分のテキストデータ (String)	2

- メソッド一覧

メソッド名	機能
ファイル名を指定してテキストデータを読み込む (String)	引数で指定した名前のファイルを開いてテキストを読み込み、ファイルを閉じる
ファイル名を指定して開く (String)	引数で指定した名前のファイルを開く (別途閉じる必要有り)
ファイルを閉じる ()	開いているファイルを閉じる (開いていない場合は無効)
開いているファイルから一行ずつ読み込む ()	開いているファイルから一行分のテキストを読み込む (ファイルは開いたまま)
開いているファイルから一単語ずつ読み込む ()	開いているファイルから一単語分のテキストを読み込む (ファイルは開いたまま)

- テキストファイル出力

- イベント：なし
- メソッド一覧

メソッド名	機能
ファイル名を指定して文字列を書き出す (String, String)	第一引数で指定した名前のファイルに第二引数で指定したテキストを書き出す

- 標準入力

- データ生成イベント

イベント発生条件	内包データ	イベント番号
標準入力からテキストデータ読み込み	入力されたテキストデータ (String)	0

- メソッド一覧

メソッド名	機能
Enter キー入力まで文字列を読み込む ()	標準入力 (コンソール) からのキー入力を開始し、Enter キーが押された時点でそれまでの入力をテキストとして読み込む

- 標準出力

- イベント：なし
- メソッド一覧

メソッド名	機能
文字列を出力する (String)	引数で指定したテキストを標準出力 (コンソール) に書き出す
文字列を出力して改行する (String)	引数で指定したテキストを標準出力 (コンソール) に書き出した後に改行する
改行する ()	標準出力 (コンソール) の表示を改行する

- 標準エラー出力

- イベント：なし
- メソッド一覧

メソッド名	機能
文字列を出力する (String)	引数で指定したテキストを標準エラー出力 (コンソール) に書き出す

- 時間計測
 - イベント：なし
 - メソッド一覧

メソッド名	機能
タイトルを指定して計測を開始する(String)	引数で指定したテキストをタイトルに設定して計測を開始(終了まで計測時間が累積していく)
計測を終了する()	計測開始からの時間をタイトルと合わせたテキストとして戻り値で返す

4. コンポーネント使用例

4.1. サンプルアプリケーション概要と操作方法

この文書で説明するコンポーネントの使用例として、サンプルアプリケーション（AP_DATA¥Sample¥テキスト入出力と時間計測.mzax）が用意されています。このサンプルは、テキストファイルを読み込んでテキストエリアに表示し、テキストエリアの内容をテキストファイルとして保存するアプリケーションです。

このサンプルを実行するには、コンソールありのビルダーまたはローダーでアプリケーションデータをロードする必要があります。これは、標準入力や標準出力がコンソールに設定されるからです。コンソールありのビルダーを起動する場合には、図 1 に示すようにスタートメニューから「アプリケーションビルダー(コンソール)」を選びます。

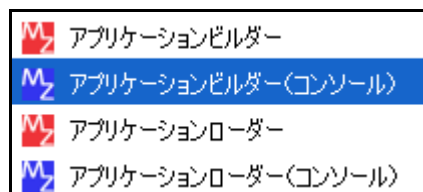


図 1 コンソールありのビルダー起動メニュー

前述のアプリケーションデータをビルダーでロードして実行すると、図 2 に示すフレームが表示されます。

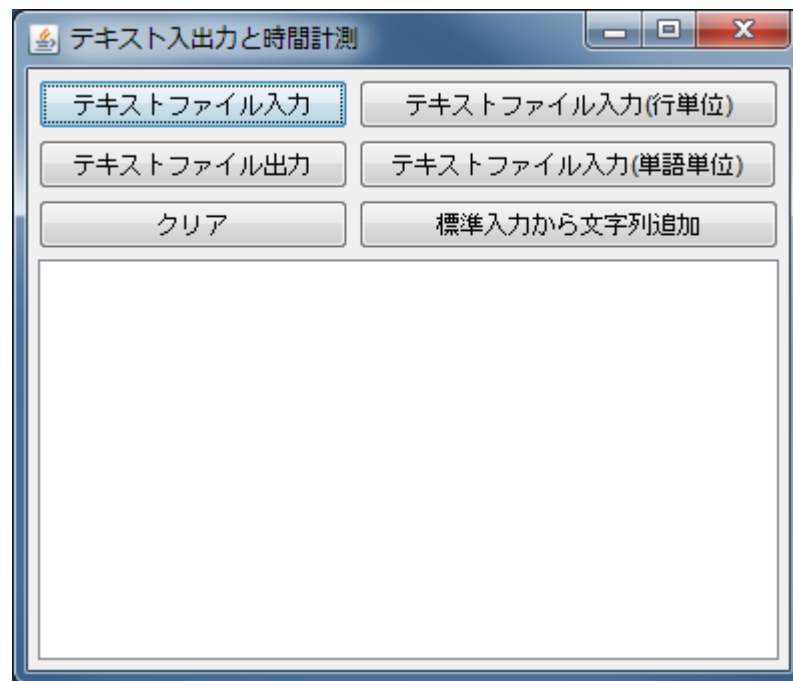


図 2 サンプルアプリケーション実行画面

各ボタンの機能は次の通りです。

- 「テキストファイル入力」:
ファイル選択ダイアログが表示され、テキストファイルを選択するとその内容がテキストエリアに表示されます。
- 「テキストファイル出力」:
ファイル選択ダイアログが表示され、ファイル名を指定するとテキストエリアに表示されている内容をそのファイルに保存します。
- 「クリア」:
テキストエリアに表示されている内容がすべて消去されます。

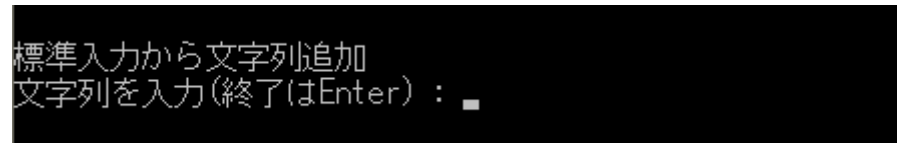
- 「テキストファイル入力(行単位)」および「テキストファイル入力(単語単位)」：
「テキストファイル入力」ボタンと同様にファイル選択ダイアログが表示され、テキストファイルを選択するとその行数または単語数をカウントして、その結果をダイアログで表示します。(ただし、ここで言う単語とは、スペースまたはタブで区切られた文字列のまとまりを指します。)
- 「標準入力から文字列追加」：
コンソールからのキー入力を受け付ける状態になり、**Enter** キーを押した時点でそれまでの入力をテキストとして読み込んで、テキストエリアの表示に追加します。

ボタン以外の機能は次の通りです。

- テキストエリアの内容は編集することができます。
- 各ボタンを押して処理が実行されたときに、コンソールにテキストが書き出されます。
- テキストファイル入力とテキストファイル出力を実行した場合には、その実行時間もコンソールに書き出されます。

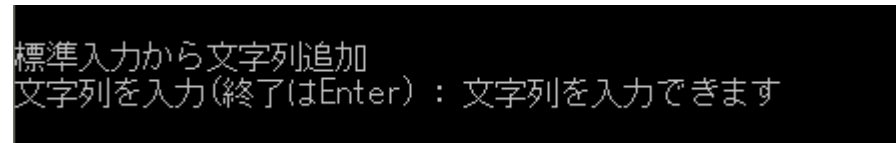
「標準入力から文字列追加」ボタンの機能に関して補足説明します。図 1 に示すメニューのコンソールありではなく、コンソールなしのビルダーやローダーでこのサンプルを実行した場合、このボタンを押すとログファイルにエラーが出力されます。

正しくコンソールありで実行した場合は、このボタンを押すと図 3に示すようにコンソールからの入力待ち状態に入ります。一度入力待ち状態に入ると、**Enter**キー入力があるまでウィンドウ側は操作できなくなります。



標準入力から文字列追加
文字列を入力(終了はEnter) : █

図 3 コンソールからの入力待ち状態



標準入力から文字列追加
文字列を入力(終了はEnter) : 文字列を入力できます

図 4 コンソールからのテキスト入力例

コンソールウィンドウにフォーカスを当て、図 4 に示すようにテキストを入力して**Enter**キーを押すと、入力したテキストが図 5 のようにテキストエリアに表示されます。

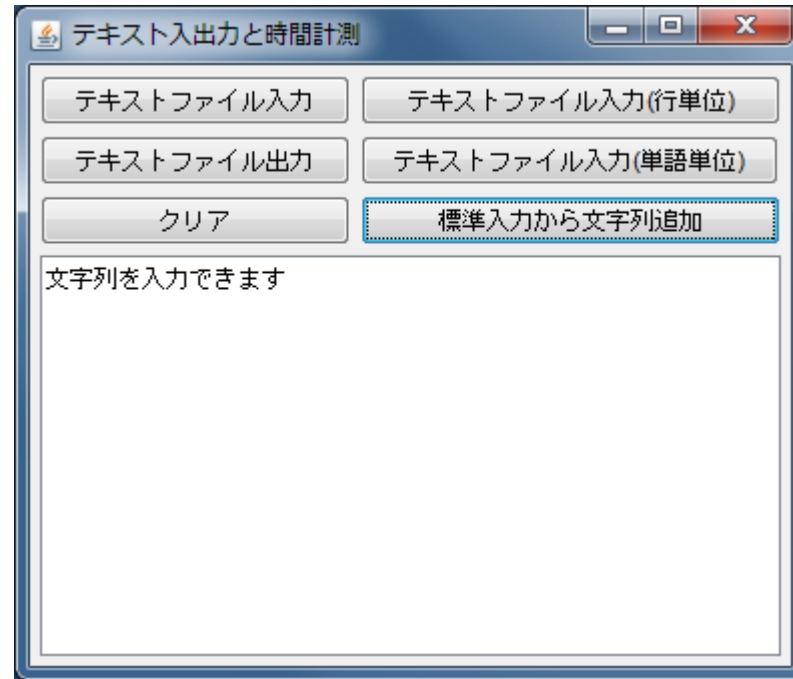


図 5 コンソールからのテキスト入力例の結果

4.2. サンプルアプリケーションに含まれる各コンポーネントの使用方法

上で示したサンプルアプリケーションの設計図をみながら、この文書で扱うコンポーネントの使用方法について説明します。

注意事項：この文書の冒頭で示した以外のコンポーネント（例えばボタンやテキストエリア）に関しては、使用法がわかっているものとして説明を省きます。それらに関しては別途チュートリアルやリファレンス、*Javadoc* ドキュメントを参照してください。

4.2.1. 標準出力

「テキストファイル入力」ボタンを押したときの処理を例に、標準出力の使い方を示します。ビルダー上に標準出力を追加して、出力を行いたい場所で図

のように標準出力のメソッドを起動します。複数の場所で出力を行う場合も、コンポーネントは一つだけあれば充分です。

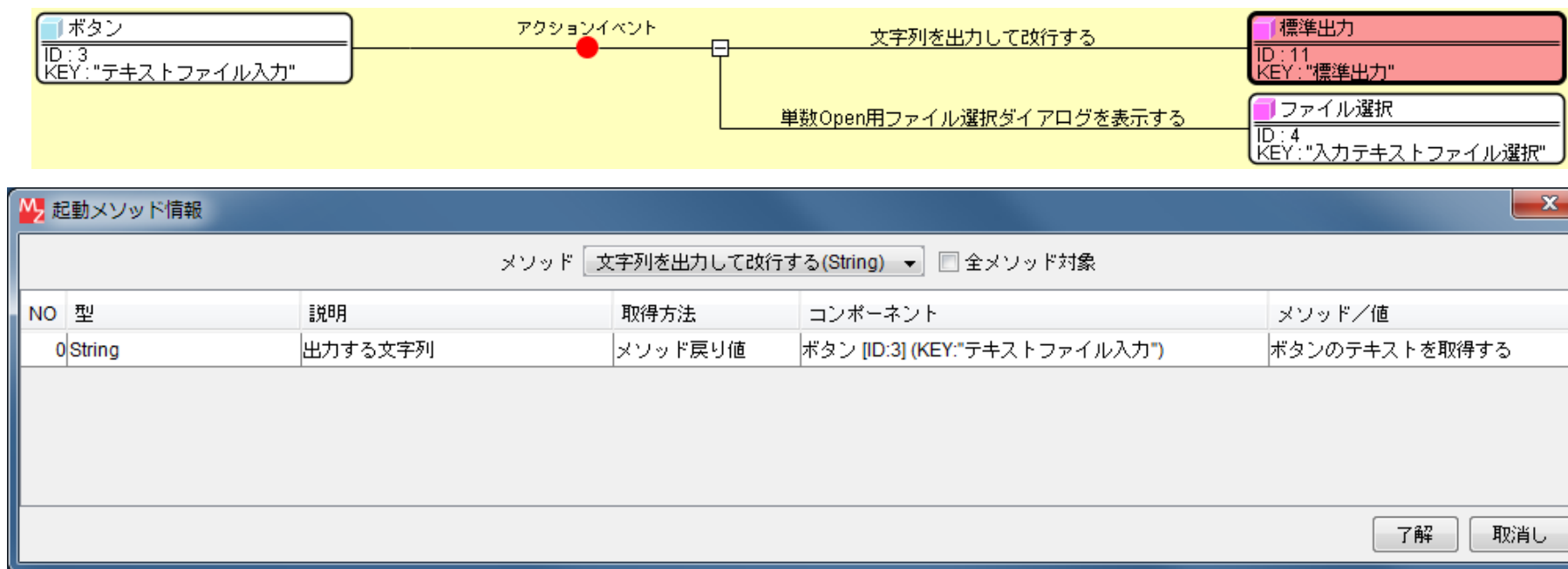
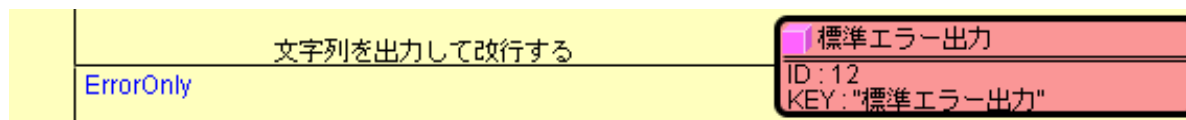


図 6 標準出力の使用例

ここではメソッド「文字列を出力して改行する(String)」を起動して、押したボタンに表示されているテキスト「テキストファイル入力」を出力しています。他にメソッド「文字列を出力する(String)」もあり、こちらを起動すると改行なしで出力することも可能です。

4.2.2. 標準エラー出力

標準エラー出力は、基本的に標準出力と使用方法は変わりません。エラー出力であることを明示する場合に使用します。このサンプルでは、メソッド起動のエラー処理として標準エラー出力を使用しています。この例では何かエラーが生じたときのみ「エラー」というテキストがコンソールに出力されます。



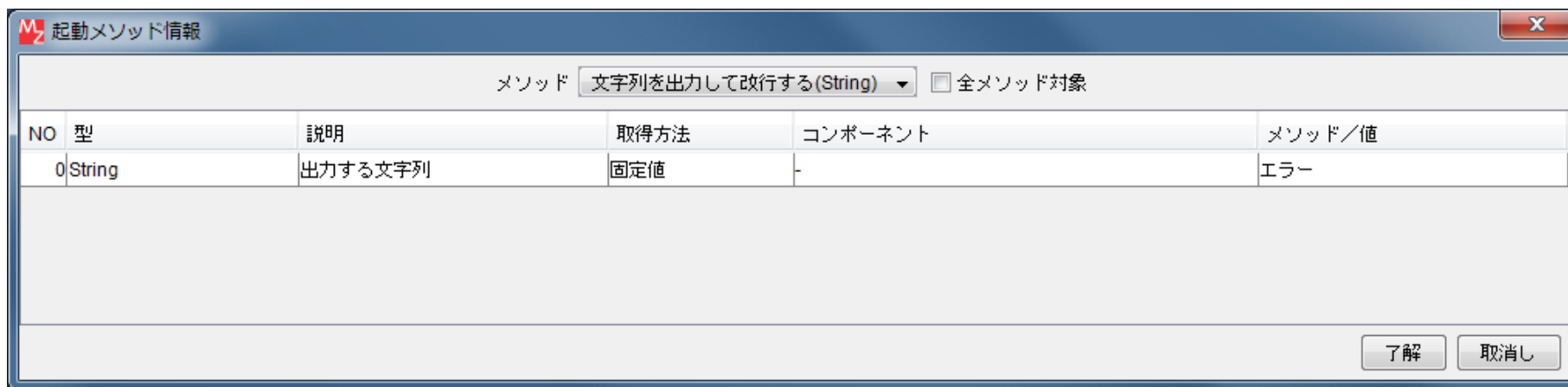


図 7 標準エラー出力の使用例

4.2.3. 標準入力

「標準入力から文字列追加」ボタンを押したときの処理を例に、標準入力の使用方法を説明します。標準入力は、**Enter** キーが押されるまでに入力された文字列を読み込み、データ生成イベントを発生させます。このイベントには入力された文字列がイベント内包データとして含まれます。このサンプルでは、データ生成イベントが発生したときに、テキストエリアにイベントに内包された文字列を追加するという処理を行っています。

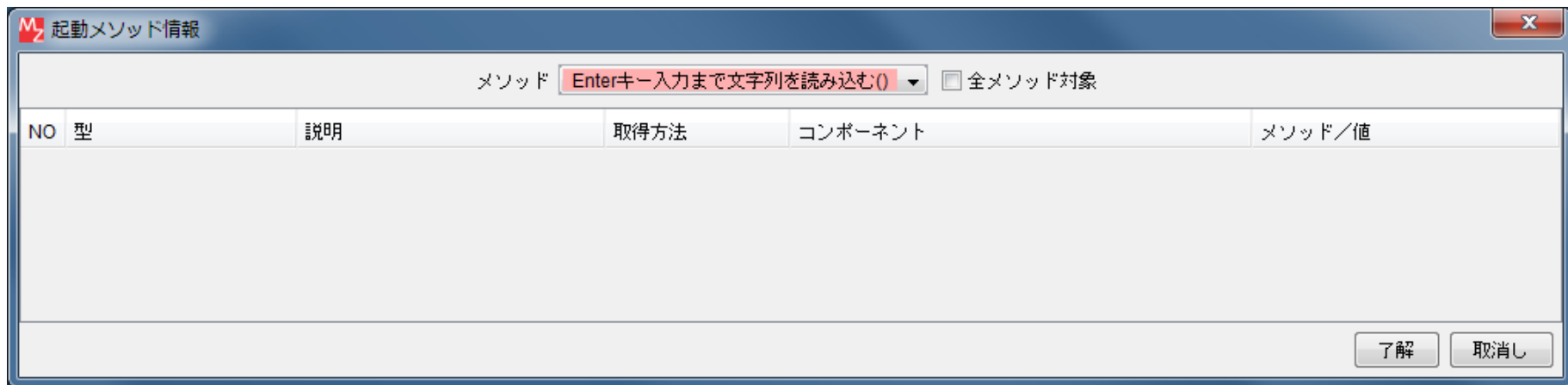
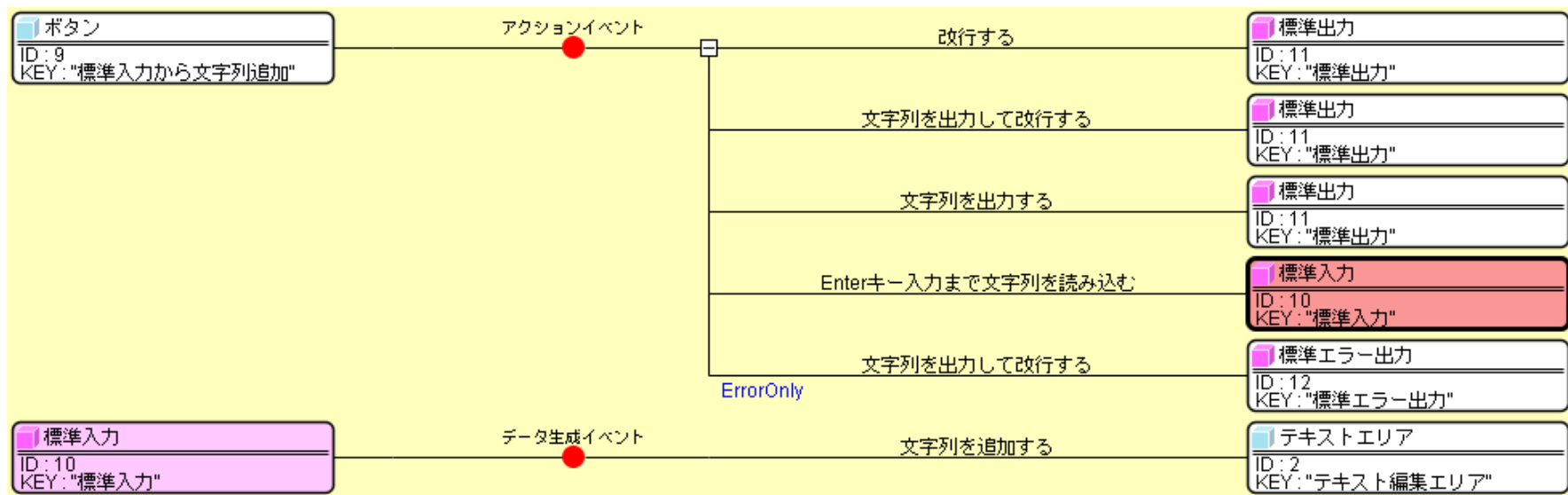


図 8 標準入力の使用例

4.2.4. 時間計測

「テキストファイル入力」ボタンを押して表示されたダイアログでファイルを選択したときの処理を例に、時間計測の使い方を説明します。実際にテキストファイルからデータを読み込む処理は3番目のメソッド起動で行われており、その直前(2番目)で時間計測の「タイトルを指定して計測を開始する」というメソッドを起動しています。

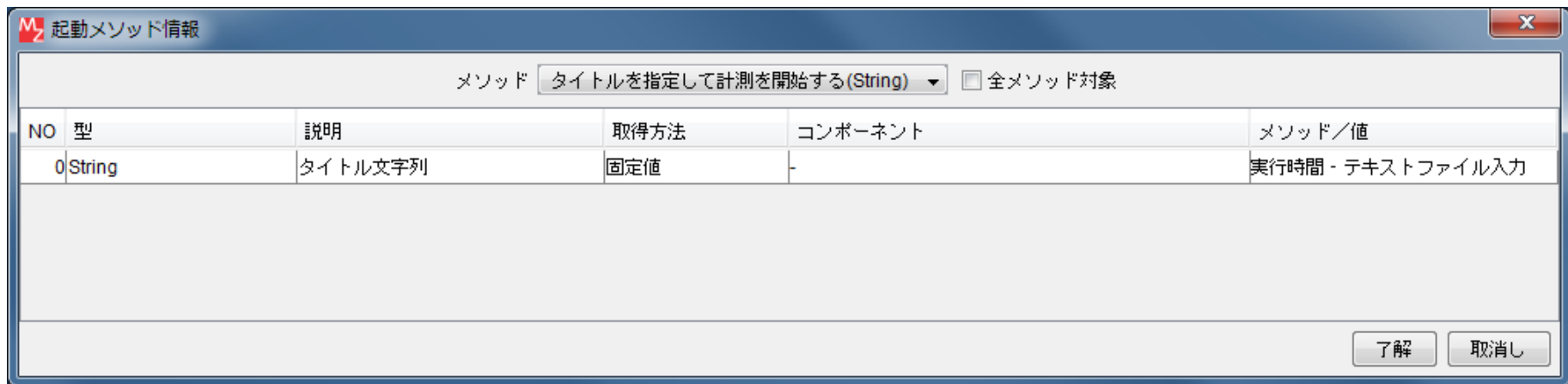
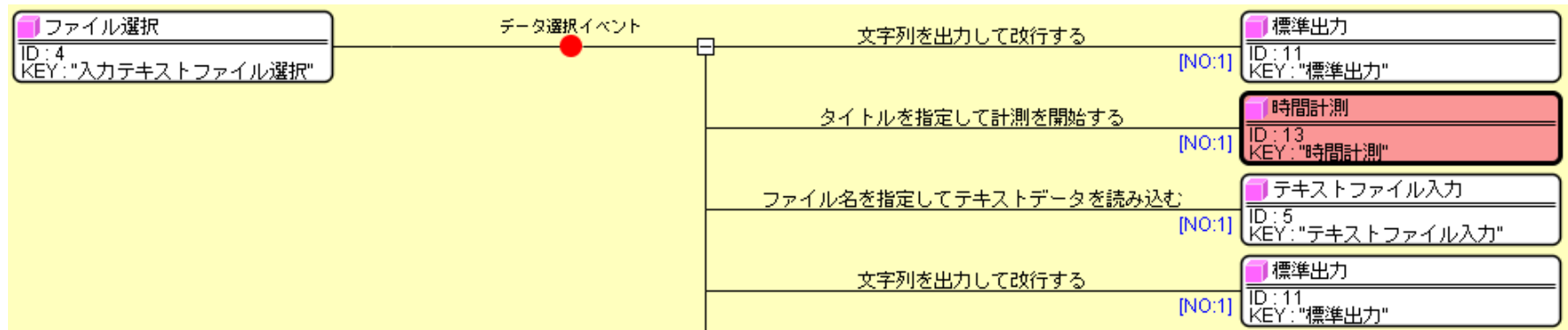
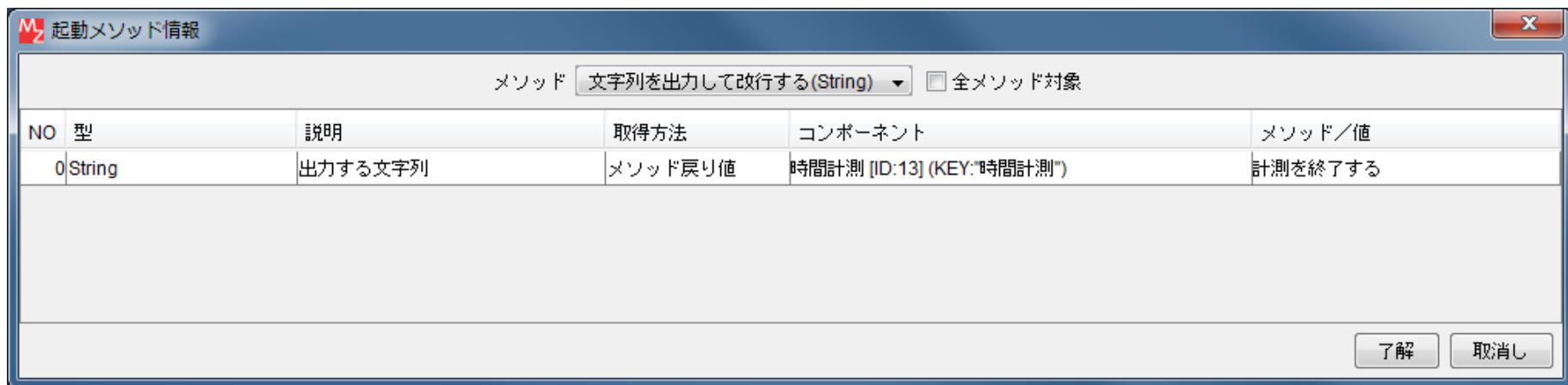
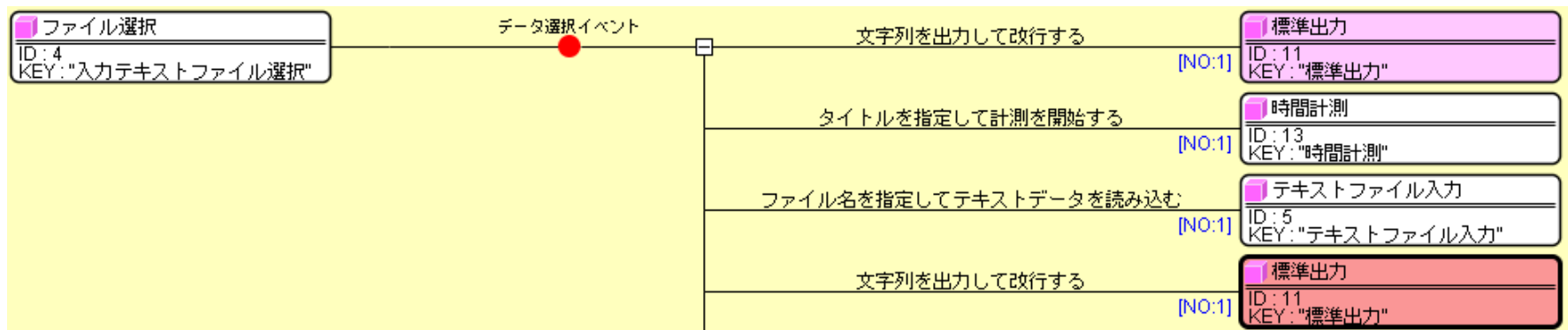


図 9 時間計測の使用例：計測開始

続いて、実際の処理の直後(4番目)で標準出力に文字列を出力しています。このとき出力する文字列として、時間計測の「計測を終了する」というメソッドの戻り値を指定しています。



```

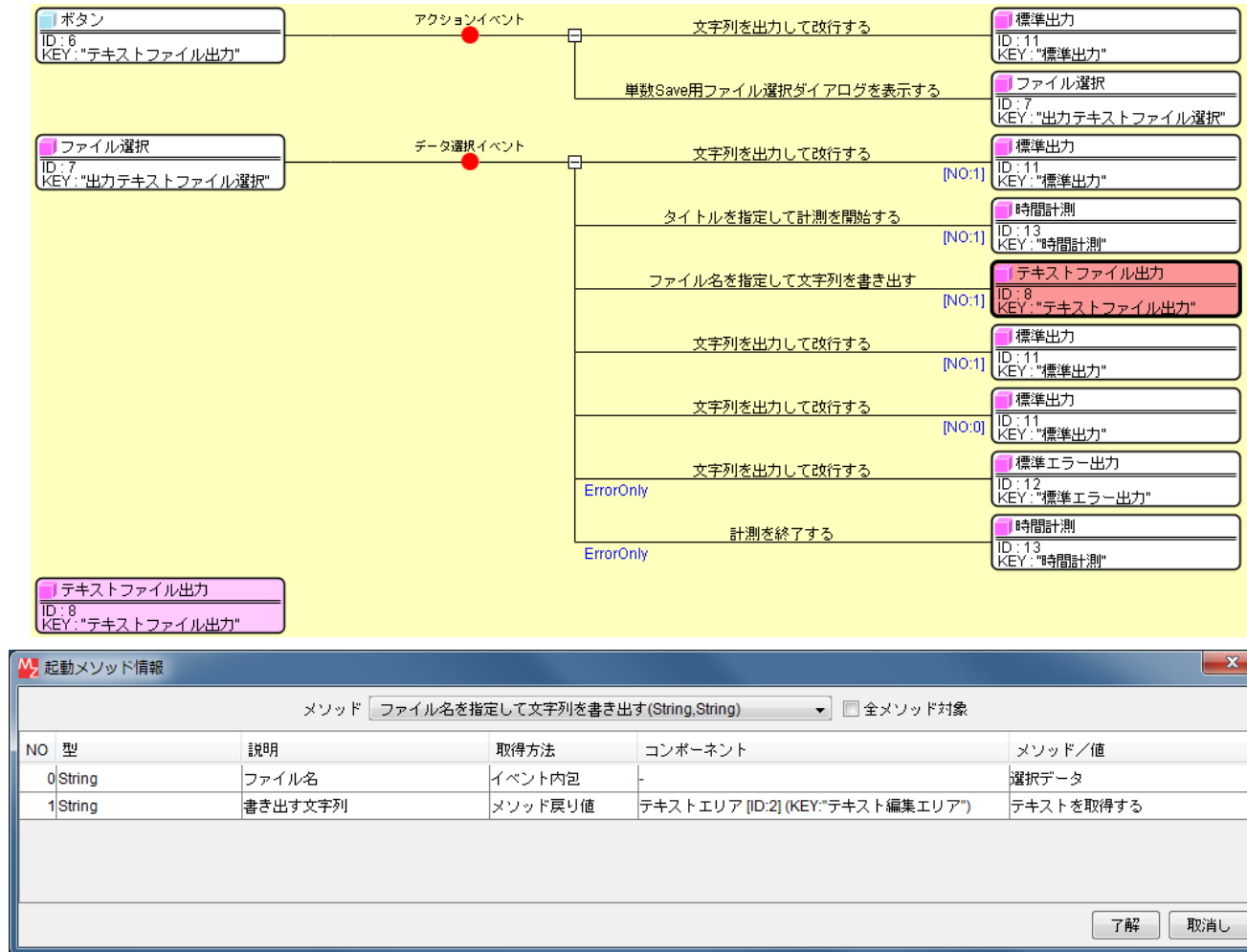
実行 - テキストファイル入力
テキストエリアに設定
実行時間 - テキストファイル入力 : 2640 msec
  
```

図 10 時間計測の使用例：計測終了

このように、対象とする処理の前後で、時間計測のメソッドを起動して計測の開始と終了を行うと、処理に要した時間を計測することができます。計測結果は終了メソッド呼び出し時に文字列の戻り値として得られます。

4.2.5. テキストファイル出力

「テキストファイル出力」ボタンを押したときの処理を例に、テキストファイル出力の使用法を説明します。ボタンを押すとダイアログが表示され、ファイルが選択されたらデータ選択イベントが発生するので、テキストファイル出力のメソッド「ファイル名を指定して文字列を書き出す」を起動しています。



起動メソッド情報

メソッド: ファイル名を指定して文字列を書き出す(String,String) 全メソッド対象

NO	型	説明	取得方法	コンポーネント	メソッド/値
0	String	ファイル名	イベント内包	-	選択データ
1	String	書き出す文字列	メソッド戻り値	テキストエリア [ID:2] (KEY:"テキスト編集エリア")	テキストを取得する

了解 取消し

図 11 テキストファイル出力の使用例

4.2.6. テキストファイル入力

「テキストファイル入力」ボタンを押したときの処理を例にテキストファイル入力の使用法を説明します。ボタンを押すとダイアログが表示され、ファイルが選択されたらデータ選択イベントが発生するので、テキストファイル入力のメソッド「ファイル名を指定してテキストデータを読み込む」を起動しています。

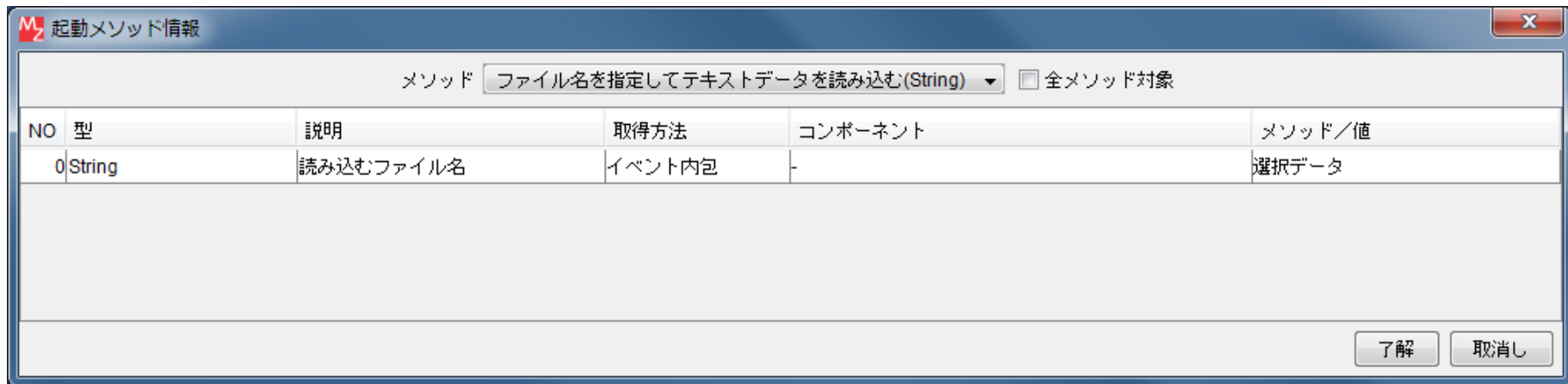
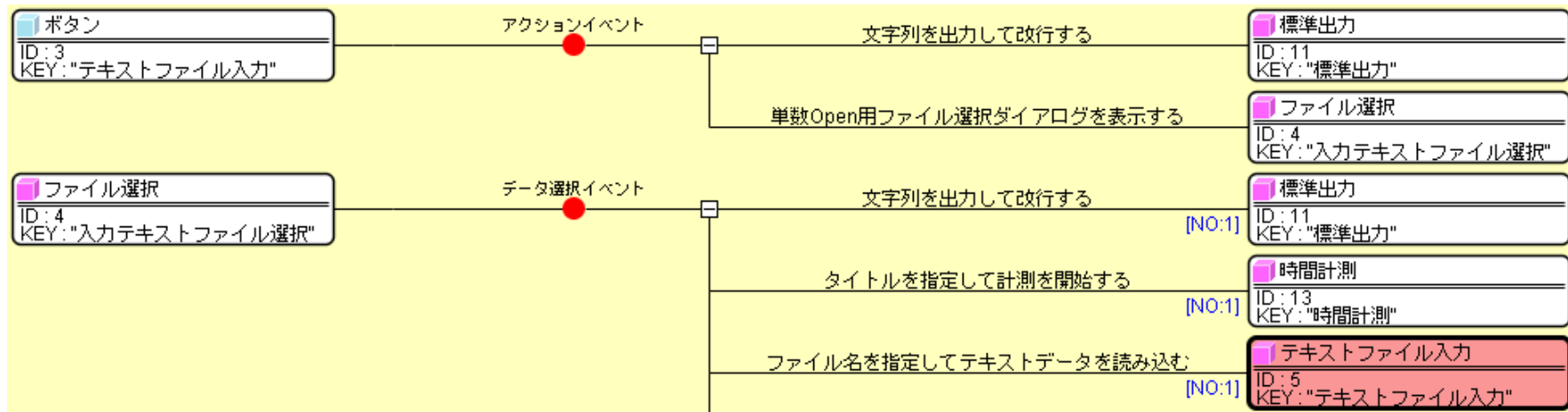


図 12 テキストファイル入力の使用例：テキストデータの読み込み

続いて、テキストデータが正常に読み込まれた際には、テキストファイル入力からデータ生成イベントが発生し、そのテキストデータはイベント内包データとして取得することができます。ここでは、イベントに内包されたテキストデータをテキストエリアに設定する処理を行っています。

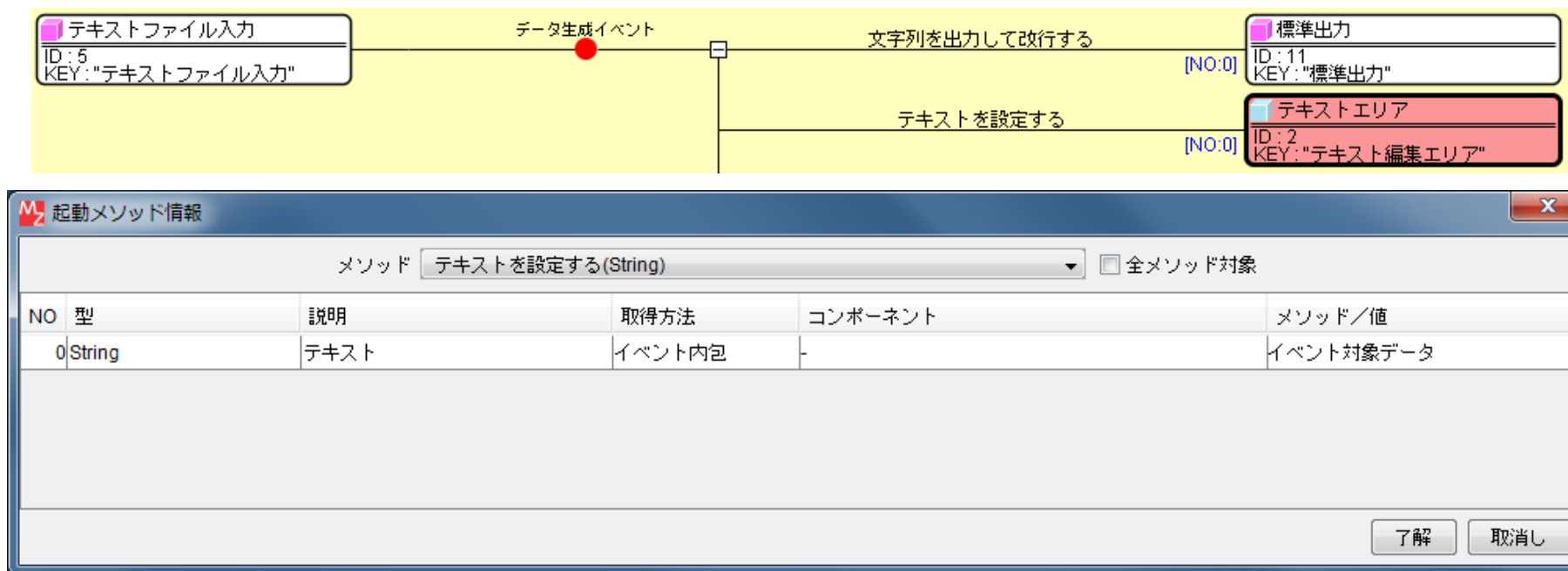


図 13 テキストファイル入力の使用例：入力テキストの表示

テキストファイル入力には、前述のようにファイルから一括でテキストデータを読み込む他に、ファイルから行単位で読み込む機能と単語単位で読み込む機能があります。以下の図は、「テキストファイル入力(行単位)」ボタンを押した場合の処理を示しています。ファイル選択ダイアログが開いてテキストファイルを選択した後、一括で読み込む場合とは異なり、実際に読み込む前後でファイルを開く処理と閉じる処理が必要になります。つまり、一連の処理として、テキストファイル入力のメソッド起動を3つ続けて行い、起動するメソッドはそれぞれ「ファイル名を指定して開く」、「開いているファイルから1行ずつ読み込む」、「ファイルを閉じる」となります。開く処理を行わずに読み込んだ場合はエラーが発生し、閉じる処理をしなかった場合は処理に不都合が生じる場合があるので、ご注意ください。

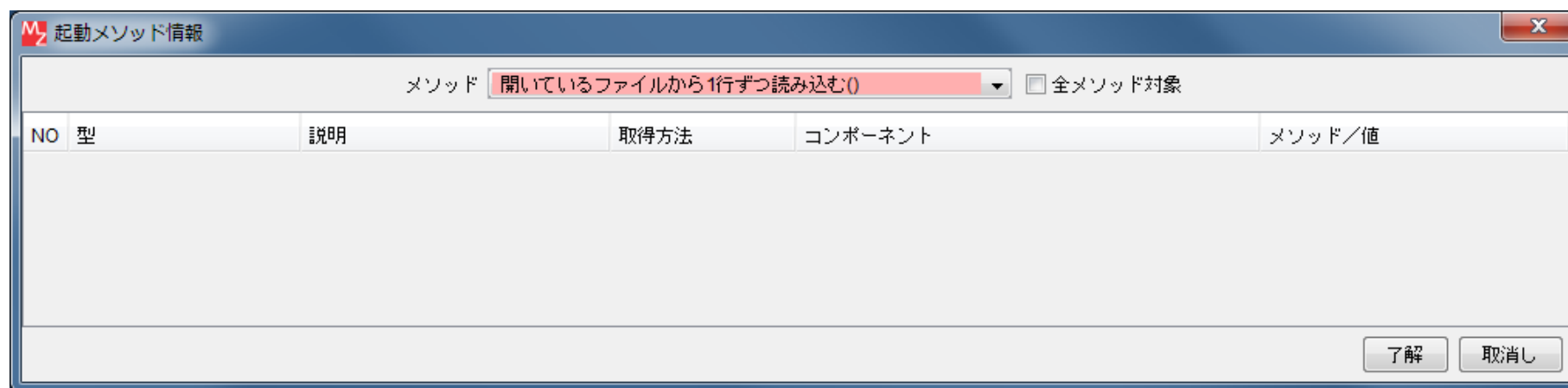
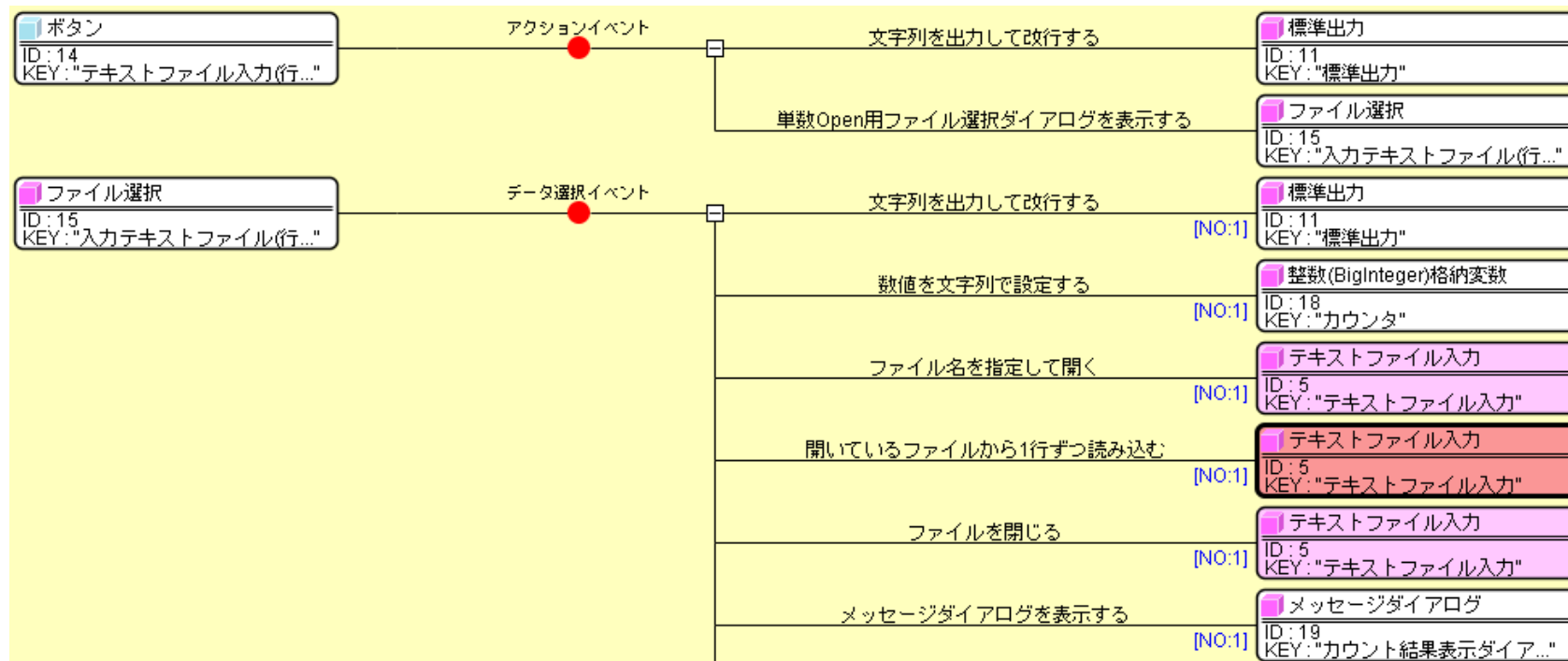


図 14 テキストファイル入力の使用例：行単位の読み込み

テキストファイル入力はデータを読み込むとデータ生成イベントが発生しますが、このとき読み込み方法によってイベント番号が異なります。一括で読み込む場合はイベント番号が 0 となり、行単位で読み込む場合は 1、単語単位で読み込む場合は 2 となります。ここでは、一括読み込みの場合は前述のようにテキストエリアにテキストを設定し、行単位読み込みと単語単位読み込みの場合は読み込み数をカウントする処理をしています。この例からもわかるように、行単位や単語単位の読込では、読み込み回数分のデータ生成イベントが発生します。

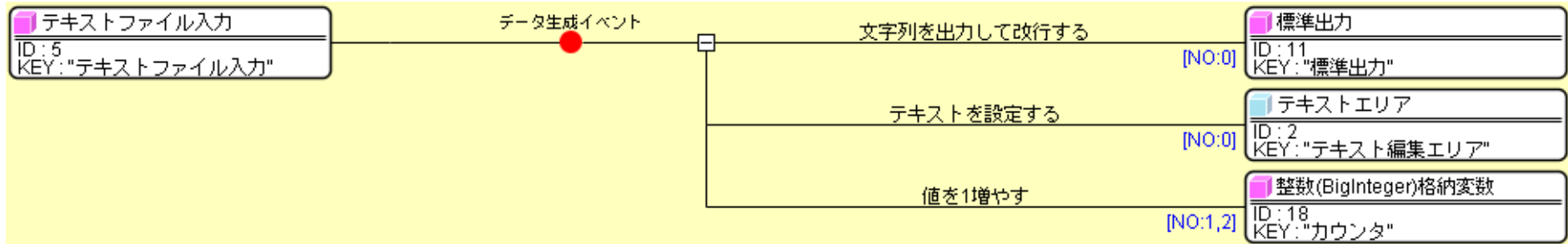


図 15 テキストファイル入力のデータ生成イベントとイベント番号

また、イベント内包データにはそのとき読み込んだデータが含まれます。サンプルアプリケーションでは行単位と単語単位で読み込んだときはそのデータを使用しない処理をしていますが、実際には行単位の場合は一行分のテキスト、単語単位の場合にはスペースやタブで区切られた文字列の一まとまりを使用することができます。