

Installation and Operation Guide

NAVIAEO v.1.5

インストール&機能ガイド

```
"Shift-JIS" ?>
no" Software="HyperMesh" Version="6.0">
J Name="Reading in CAD data" Tip="Select a type of model to read in." MenuID="model">
  <GUIDE>guide\model\model.html</GUIDE>
  <MENU Name="IGES Model" Tip="Read in IGES file" MenuID="model 1">
    <PROGRAM Lang="Peri" PrgFile="file_find_igs.pl" CallFunc="main_func"/>
    <CMD Type="ok can">
      <SET Variable="igs_Name" Type="string" IDValue="igsName"/>
      <PROGRAM Lang="Peri" PrgFile="File_Read.pl" CallFunc="main_func"/>
      <COMBO CtrlID="igsName"
        <ITEMCREATE>
          <ITEM Label="Name of the file to read in" ID="name_of_the_file_to_read_in">
            <GET Variable="igs_list"/>
            </ITEMCREATE>
          <PROGRAM Lang="Peri" PrgFile="File_Read.pl" CallFunc="main_func"/>
          <CMD Type="ok can">
            <SET Variable="igs_Name" Type="string" IDValue="igsName"/>
          </PROGRAM>
        </COMBO>
      </MENU>
    <MENU Name="SAT Model" Tip="Read in SAT file" MenuID="model 2">
      <PROGRAM Lang="Peri" PrgFile="File_Read.pl" CallFunc="main_func"/>
      <COMBO CtrlID="igsName"
        <ITEMCREATE>
          <ITEM Label="Name of the file to read in" ID="name_of_the_file_to_read_in">
            <GET Variable="igs_list"/>
            </ITEMCREATE>
          <PROGRAM Lang="Peri" PrgFile="File_Read.pl" CallFunc="main_func"/>
          <CMD Type="ok can">
            <SET Variable="igs_Name" Type="string" IDValue="igsName"/>
          </PROGRAM>
        </COMBO>
      </MENU>
    <MENU Name="Create Mesh" Tip="Apply meshing to the model" MenuID="mesh">
      <GUIDE>guide\mesh\mesh.html</GUIDE>
      <KNOWHOW>C:\user\NAVIAEO\knowhow\mesh.html</KNOWHOW>
      <MENU Name="Create Mesh" Tip="Meshing will be applied by inputting of average length of element"
        <CMD Type="ok can">
          <MENU>
            <PROGRAM>0902</PROGRAM>
            <MENU Name="Create Mesh" Tip="Apply meshing to the model" MenuID="mesh 1">
              <EDIT CtrlID="m_value" Label="Overall average length of element">10</EDIT>
              <MENU Name="Create Mesh" Tip="Meshing will be applied by inputting of average length of element" MenuID="mesh 1">
                <CMD Type="ok can">
                  <PROGRAM>0000</PROGRAM>
                </MENU>
              <MENU Name="Addition of local meshing" MenuID="mesh 2">
                <MENU Name="Apply to the edge" Tip="Local meshing will be applied to the edge." MenuID="mesh 3">
                  <CMD Type="ok can">
                    <PROGRAM>0000</PROGRAM>
                    <EDIT CtrlID="m_value" Label="Number of divisions"/>
                    <MENU Name="Apply to the edge" Tip="Local meshing will be applied to the edge." MenuID="mesh 1">
                      <CMD Type="ok can">
                        <PROGRAM>0000</PROGRAM>
                        <EDIT CtrlID="m_value" Label="Number of divisions"/>
                      </MENU>
                    </MENU>
                  </MENU>
                </MENU>
              </MENU>
            </MENU>
          </MENU>
        </MENU>
      </MENU>
    </MENU>
  </MENU>
</MENU>
```

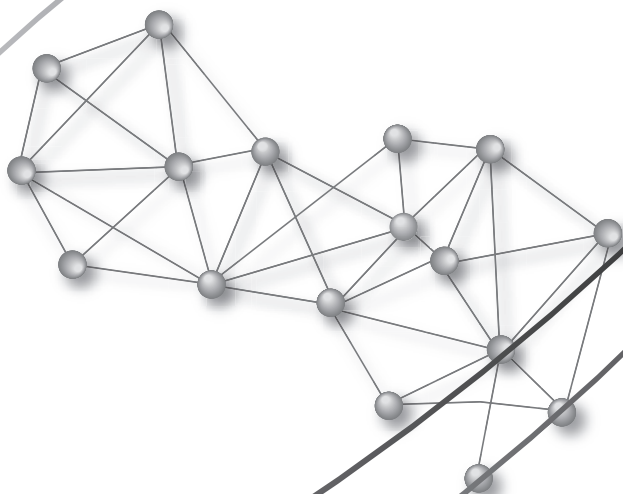
NAVIAEO

navideo



NAVIAEO v.1.5

インストール&機能ガイド



インテグラル・テクノロジー株式会社

目次

第一章	NAVIAEO のインストール	
	1. はじめに.....	1
	本書の目的.....	1
	インストールの前準備.....	1
	2. NAVIAEO のインストール.....	1
	3. NAVIAEO の概要.....	2
	NAVIAEO の起動・終了.....	2
	NAVIAEO のフォルダ構成.....	3
第二章	カスタマイズ	
	1. カスタマイズ概要.....	5
	NAVIAEO のカスタマイズ項目.....	5
	データ保持領域について.....	6
	2. xml の概要.....	8
	xml ファイルの種類.....	8
	xml ファイルの保管場所.....	8
	xml の基本.....	8
	xml ファイルの例.....	9
	3. xml タグ詳細.....	11
	<CMD>.....	11
	<COMBO>.....	13
	<EDIT>.....	14
	<FILESELECT>.....	16
	<GET>.....	18
	<GUIDE>.....	19

<INPUT>	20
<ITEMCREATE>	21
<KNOWHOW>	21
<LIST>	22
<MENU>	24
<NaviaXml>	26
<PROGRAM>	27
<ROOT>.....	28
<SET>.....	29
<STATIC>.....	30
<TREE>	31
<TREETYPE>.....	32
<!-->	34
<?xml>	35
4. Perl によるカスタマイズ.....	36
カスタマイズの概要.....	36
カスタマイズ事例	38
5. グレーアウトの設定.....	40
グレーアウト設定ファイルの保管場所.....	40
グレーアウトの設定方法	40
グレーアウトの設定例	41
6. NAVIA MANAGER.....	42
コマンド登録機能	42
7. その他.....	43
NAVIAEO の予約変数	43
NAVIAEO の予約コマンド	44

第一章 NAVIAEO のインストール

1. はじめに

本書の目的

本書は、NAVIAEO のインストール方法、使用方法、カスタマイズについて記載しています。

インストールの前準備

NAVIAEO を安定して動作させるには、ベースソフトのシステム要件を満たし、ベースソフトが正常に動作する環境が必要です。





まず、ベースソフトが起動することを確認してください。インストールされていない場合はここでインストール作業を行ってください。

2. NAVIAEO のインストール

1. NAVIAEO インストール CD を CD ドライブに挿入すると、NAVIAEO インストーラが起動します。(もし起動しない場合は、CD ドライブを開き、「Product_setup.exe」をダブルクリックします。)
2. インストーラの指示に従い、操作を進めてください。
3. 別途供給されたライセンスファイルをインストールフォルダへコピーしてください。

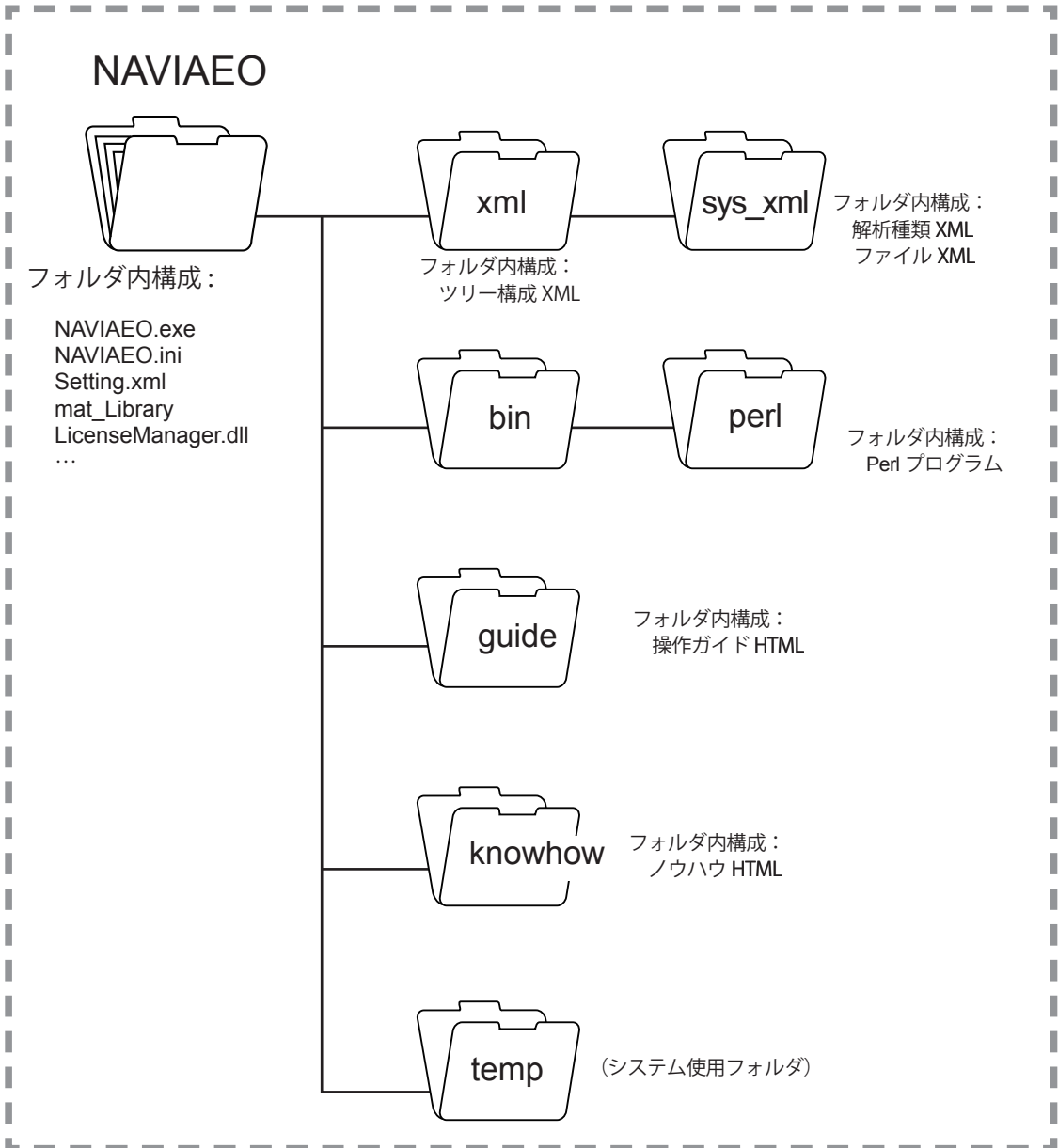
3. NAVIAEO の概要

NAVIAEO の起動・終了

1. デスクトップの「NAVIAEO アイコン」 をダブルクリックし、起動します。
2. ダイアログボックスが表示されるので、そこでソフトとサブモジュールをメニューから選択します。
3. 「ツリーセレクトアイコン」 を押し、ツリーの種類を選択します。
4. ツリーの種類を選択すると、NAVIA ツリーが表示されますので、手順に沿って作業を進めてください。
5. 必要に応じ、「ファイルアイコン」 を押し、作業を保存することができます。「名前をつけて保存」の際、拡張子は自動付与されるため不要です。（*.navi が生成されます。）
6. フォルダパス、ファイル名には半角英数字のみが使用可能です。
7. NAVIAEO の終了は「閉じるアイコン」 をクリックしてください。

NAVIAEO のフォルダ構成

NAVIAEO のフォルダ構成は次の通りです。(インストール時 knowhow フォルダは存在しません。必要に応じて作成してください。)





第二章 カスタマイズ

1. カスタマイズ概要

この章では NAVIAEO で行うカスタマイズ項目について紹介します。各カスタマイズ項目の詳細については、本章「2. xml の概要」以降で説明します。

また、「データ保持領域」についても説明します。「データ保持領域」は xml や Perl を使ってカスタマイズを行う上で重要な内容です。

NAVIAEO のカスタマイズ項目

NAVIAEO でカスタマイズ可能な項目は下記の 3 つです。

xml のカスタマイズ

NAVIAEO で表示される様々な GUI (ツリー) の変更及び実際に動作させる処理 (Perl プログラム) を指定することができます。

グレースアウトのカスタマイズ

NAVIAEO のツリー項目で選択可能、不可能の条件制御を指示させることができます。

Perl を使ったカスタマイズ

xml やコマンドでは対応できない処理を行う際に Perl を使います。NAVIAEO 内で動作させる処理を Perl で外部プログラムとして作成し、xml で指示することにより実行することができます。

データ保持領域について

データ保持領域とは

- データ保持領域とは、NAVIAEO の xml 内において共通の変数を定義する際に使用する領域のことです。変数をデータ保持領域に定義するとその変数のデータは、NAVIAEO の実行中は保持されます。NAVIAEO の作業状況をファイルに保存すると、このデータも同じファイルに保存され、次回このファイルを使って NAVIAEO の作業を開始する際には保存されたデータが有効になります。
- 変数をデータ保持領域に定義する必要があるのは、下記の 2 つの場合です。

1. <CMD> タグ内で使用した変数のデータを別の <CMD> タグ内でも使用する場合

データ保持領域に定義しない場合、その変数のデータは、定義した <CMD> タグ内のみで有効であり、それ以外の <CMD> タグ内では無効となります。したがって、別の <CMD> タグ内でもその変数を使用したい場合、データ保持領域に定義する必要があります。

2. Perl 及び xml 内で使用した変数のデータを Perl プログラムとの入出力に使用する場合

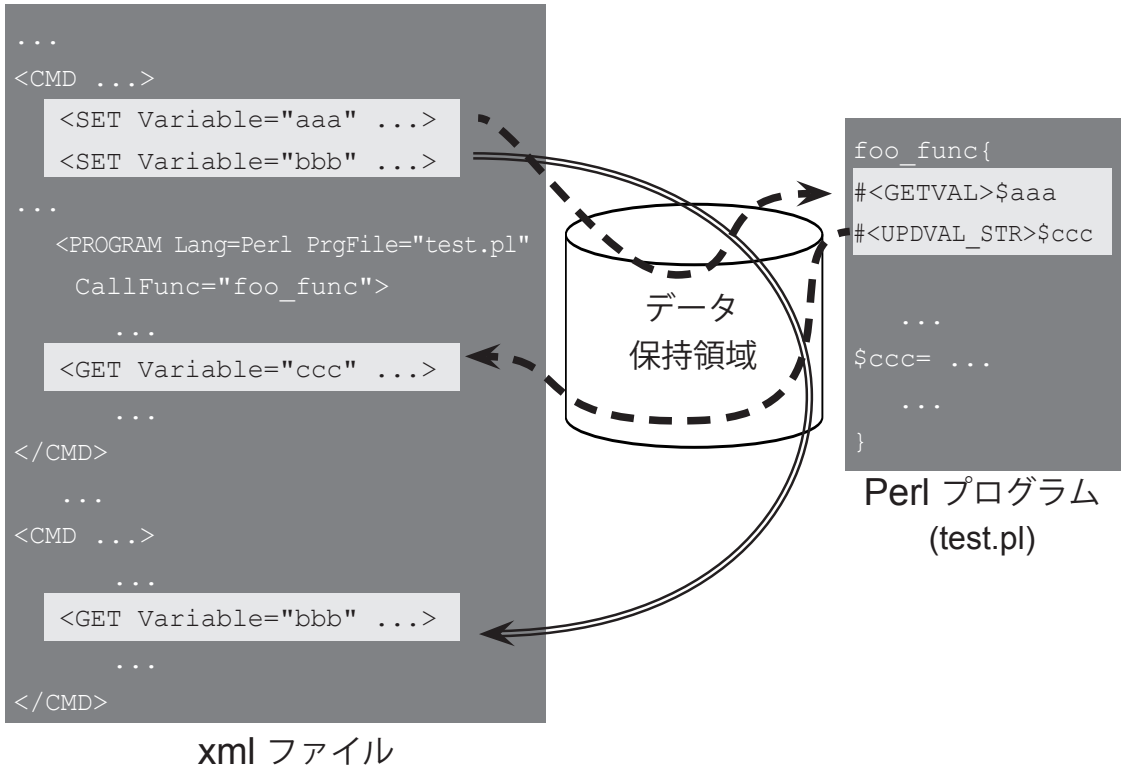
ユーザー作成の Perl プログラムと xml との間でデータをやり取りする場合は、必ずデータ保持領域に定義した変数を使う必要があります。

変数の定義方法

- xml 内でデータ保持領域に変数を定義する場合は <SET> タグを使用します。変数の値を取得する場合は <GET> タグを使用します。詳細については本章の「3. xml タグ詳細」をご覧ください。
- Perl 内でデータ保持領域に変数を定義する場合は #<UPDVAL_STR> 又は #<SETVAL_STR> を使用します。変数の値を取得する場合は #<GETVAL> を使用します。詳細については本章の「4. Perl によるカスタマイズ」をご覧ください。

データ保持領域の使用例

3つの変数 aaa、bbb、ccc をデータ保持領域に定義し、aaa と ccc で Perl プログラムとの入出力を行い、bbb の値を別の <CMD> タグ内で取得します。



2. xml の概要

ツリーの表示内容や階層構造の変更は xml ファイルの編集で行います。ここでは NAVIAEO で使用する xml の概要について説明します。

xml ファイルの種類

NAVIAEO で使用する xml ファイルには下記の 2 種類があります。

1. ツリーセレクト xml ファイル

ツリーのタイプを選択するためのファイルです。使用可能な xml タグは下記の 5 つです。

<ROOT>, <MENU>, <TREETYPE>, <?xml>, <!-->

2. NAVIA ツリー xml ファイル

選択されたツリーについての実際の作業内容を記述したファイルです。下記の 2 つの xml タグ以外が使用可能です。

<ROOT>, <TREETYPE>

(※各 xml タグの内容については、次セクション「3. xml タグ詳細」をご覧ください。)

xml ファイルの保管場所

xml ファイルは下記のフォルダへ保存してください。

- ツリーセレクト xml ファイル：<NAVIAEO インストールフォルダ>%xml%sys_xml
- NAVIA ツリー xml ファイル：<NAVIAEO インストールフォルダ>%xml

ファイルの拡張子は .xml です。なお、ツリーセレクト xml ファイル名は、AnaType_mgr.xml に固定されています。

xml の基本

xml は開始タグ (< ... >) と終了タグ (</ ... >) をセットで使用します。タグの下位項目が無い場合は、開始タグ内末尾にスラッシュ (/) を書くことで終了タグを省略することができます。なお、xml ではタグ名、属性名の大文字、小文字を区別しますのでご注意ください。

例) <SAMPLE Attribute="test"> サンプル </SAMPLE>
 <SAMPLE Attribute="test"/>

xml ファイルの例

下記に NAVIA ツリー用 xml ファイルの記述例を示します。各 xml タグの内容については、本章の「3. xml タグ詳細」をご覧ください。

XML ファイルサンプル

```

<?xml version="1.0" encoding="Shift-JIS" ?> ..... 1
<NaviaXml> ..... 2
  <TREE TreeID="Stress" Software="ooo" Version="ooo"> ..... 3
    <MENU ..... 4
      Name=" 拘束条件の設定 " ..... 5
      Tip=" 拘束条件を設定します。 " ..... 6
      <MenuID="koso"> ..... 7
        <GUIDE>guide%fix%fix1.html</GUIDE> ..... 8
        <MENU Name=" 面固定 " ..... 9
          Tip=" 面を固定します。 "
          MenuID="koso_1">
            <GUIDE>guide%fix%men_fix.html</GUIDE>
            <CMD Type="ok"> ..... 10
              <PROGRAM Lang="Perl" PrgramFile="Test.pl" CallFunc="main_func"/>
            </CMD>
          </MENU>
          <MENU Name=" 辺固定 " ..... 11
            Tip=" 辺を固定します。 "
            MenuID="koso_2">
              <GUIDE>guide%fix%hen_fix.html</GUIDE>
              <MENU Name=" 辺 ID 表示 "
                Tip=" 辺の ID を表示します。 "
                MenuID="koso_21">
                  <GUIDE>guide%fix%hen_ID.html</GUIDE>
                  <CMD Type="ok">
                    <PROGRAM>0000</PROGRAM> ..... 12
                    <EDIT CtrlID="edge1" Label=" 辺 ID"/>
                  </CMD>
                </MENU>
              </MENU>
            </MENU>
          </MENU>
        </TREE>
      </NaviaXml>
  
```

- 1** xml の宣言文 (必須)
- 2** NAVIAEO 用 xml の開始宣言タグ (必須)
- 3** ツリー ID の指定 (必須)
- 4** ツリー階層を生成します
- 5** ツリーに表示される項目名を指定します
- 6** ツールチップに表示される内容を指定します
- 7** 各メニューに他と重複しない ID を与えます (グレーアウトで使用)
- 8** 表示する操作手順ガイドを指定します
- 9** 上位タグに挟まれた MENU タグは下層メニューとなります
- 10** コマンドやボタンタイプを設定します
- 11** OK ボタンで動作させる外部プログラムを指定します
- 12** OK ボタンで動作させるコマンドを指定します

3. xml タグ詳細

ここでは各 xml タグの詳細な内容を説明します。リファレンスマニュアルとしてご利用ください。

<CMD>

概要

形式	<CMD Type= ... > ... </CMD>
包含可能要素	<COMBO>, <EDIT>, <FILESELECT>, <GUIDE>, <INPUT>, <LIST>, <PROGRAM>, <SET>, <STATIC>

説明

- <PROGRAM> タグで指定されるプログラムを実行する／しないのボタンを表示します。

属性

属性名	内容	型	必須
Type	コマンドボタンのタイプ（下記のいずれかを指定）： none, ok, ok_can, conf_ok_can, apply, apply_can, conf_apply_can	STR	○
Msg_APPLY	適用ボタンをクリックした場合に表示される内容	STR	×
Msg_OK	OK ボタンをクリックした場合に表示される内容	STR	×

コマンドボタンのタイプについて

タイプ	内容
none	ボタンを表示せず、メニュー項目選択時に直ちにプログラムを実行する
ok	OK ボタンを表示する。OK ボタンを押すとプログラムを実行し、メニューを閉じる
ok_can	OK ボタンとキャンセルボタンを表示する。OK ボタンを押すとプログラムを実行し、メニューを閉じる。キャンセルボタンを押すとプログラムを実行せずにメニューを閉じる

conf_ok_can	確認ボタン、OK ボタン及びキャンセルボタンを表示する。初期状態では確認ボタンとキャンセルボタンが有効である。確認ボタンを押した後、OK ボタンが有効になる。OK ボタンを押すとプログラムを実行し、メニューを閉じる。キャンセルボタンを押すとプログラムを実行せずにメニューを閉じる
apply	適用ボタンを表示する。適用ボタンを押すとメニューを閉じずにプログラムを実行する
apply_can	適用ボタンとキャンセルボタンを表示する。を表示する。適用ボタンを押すとメニューを閉じずにプログラムを実行する。キャンセルボタンを押すとプログラムを実行せずにメニューを閉じる
conf_apply_can	確認ボタン、適用ボタン及びキャンセルボタンを表示する。初期状態では確認ボタンとキャンセルボタンが有効である。確認ボタンを押した後、適用ボタンが有効になる。適用ボタンを押すとメニューを閉じずにプログラムを実行する。キャンセルボタンを押すとプログラムを実行せずにメニューを閉じる

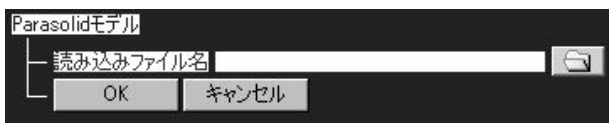
使用例

- Parasolid ファイルを読み込むコマンド (この例では 1003) を実行するメニューの場合

OK ボタンを押すと、コマンド (1003) を実行して「Parasolid モデル」メニューが閉じる。
 キャンセルボタンを押すと、何もせずに「Parasolid モデル」メニューが閉じる。

2

```
<MENU Name="Parasolid モデル " Tip=" ファイルを読み込みます "
MenuID="model_para">
  <CMD Type="ok_can">
    <PROGRAM>1003</PROGRAM>
    <FILESELECT CtrlID="f_name" Label=" 読み込みファイル名 "
      Extension="Parasolid file (*.x_t,*.x?t)|*.x_t;*.x?t|"/>
  </CMD>
</MENU>
```



<COMBO>

概要

形式	<COMBO CtrlID= ... > ... </COMBO>
包含可能要素	<GET>, <ITEMCREATE>, デフォルト値

説明

- コンボボックスを画面に表示します。固有の ID を定義し、コンボボックスで選択された項目の番号をこの ID に保存します。(デフォルト値には、初めに表示される項目がカンマ区切りリストの何番目にあたるかを指定します。)

属性

属性名	内容	型	必須
CtrlID	固有の ID	STR	○
Label	コンボボックスのラベル	STR	×
Size	コンボボックスのサイズ (下記のいずれかを指定): S: サイズ小 / M: サイズ中 (デフォルト) / L: サイズ大	STR	×
	数値: 数値によるサイズ指定 (単位は 10 ピクセル)	INT	×

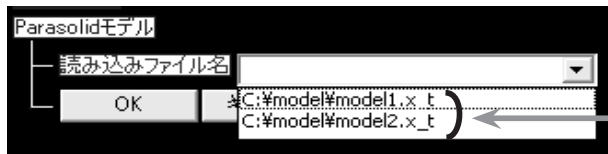
使用例

- <COMBO> タグを使って Parasolid ファイル名を取得し、インポートする場合。

```

<MENU Name="Parasolid モデル " MenuID="model_para">
  <CMD Type="ok_can">
    <PROGRAM>1003</PROGRAM>
    <COMBO CtrlID="f_name" Label=" 読み込みファイル名 ">
      <ITEMCREATE>
        C:%model%model1.x_t,C:%model%model2.x_t
      </ITEMCREATE>
    </COMBO>
  </CMD>
</MENU>

```



いずれかが選択可能

<EDIT>

概要

形式	<EDIT CtrlID= ... > ... </EDIT>
包含可能要素	<GET>, デフォルト値

説明

- パラメータ入力用のボックスを表示します。固有の ID を定義し、ボックスに入力された値をこの ID に保存します。

属性

属性名	内容	型	必須
CtrlID	固有の ID	STR	○
Label	ボックス説明	STR	×
Size	ボックスのサイズ（下記のいずれかを指定）： S: サイズ小 / M: サイズ中（デフォルト） / L: サイズ大	STR	×
	数値：数値によるサイズ指定（単位は 10 ピクセル）	INT	×

使用例

- 入力用ボックスにデフォルト値を設定する場合。

```
<EDIT CtrlID="mesh_len" Label=" 全体要素長 ">5</EDIT>
```



- 材料名と材料定数（ヤング率、ポアソン比）の入力用ボックスを表示する場合

```
<MENU Name=" 値入力 " Tip=" 材料物性値を手入力します。
  "MenuID="mat_st_man">
<GUIDE>guide¥material3.html</GUIDE>
<CMD Type="ok_can">
  <SET Variable="mat_name" Type="string" IDValue="mat_name"/>
  <SET Variable="mat_yan" Type="string" IDValue="young"/>
  <SET Variable="mat_poi" Type="string" IDValue="poisson"/>
  <PROGRAM Lang="Perl" PrgFile="comformp.pl" CallFunc="write_mat"/>
  <EDIT CtrlID="mat_name" Label=" 材料名 "/>
  <EDIT CtrlID="young" Label=" ヤング率 "/>
  <EDIT CtrlID="poisson" Label=" ポアソン比 "/>
</CMD>
</MENU>
```



<FILESELECT>

概要

形式	<FILESELECT CtrlID= ... > ... </FILESELECT>
包含可能要素	<GET>, デフォルト値

説明

- ファイル選択用又は保存用のウィンドウを表示します。固有の ID を定義し、選択されたファイル名をこの ID に保存します。
- デフォルト値としてはパスを入力します。その指定されたパスがファイル選択時の最初に表示される画面になります。入力ボックスに直接ファイル名を入力すると、最初のファイル選択画面と同じパスが自動的に付与されます。(※デフォルト値の設定がない場合は、以前の指定したパスが記憶され、次回に反映されます。)

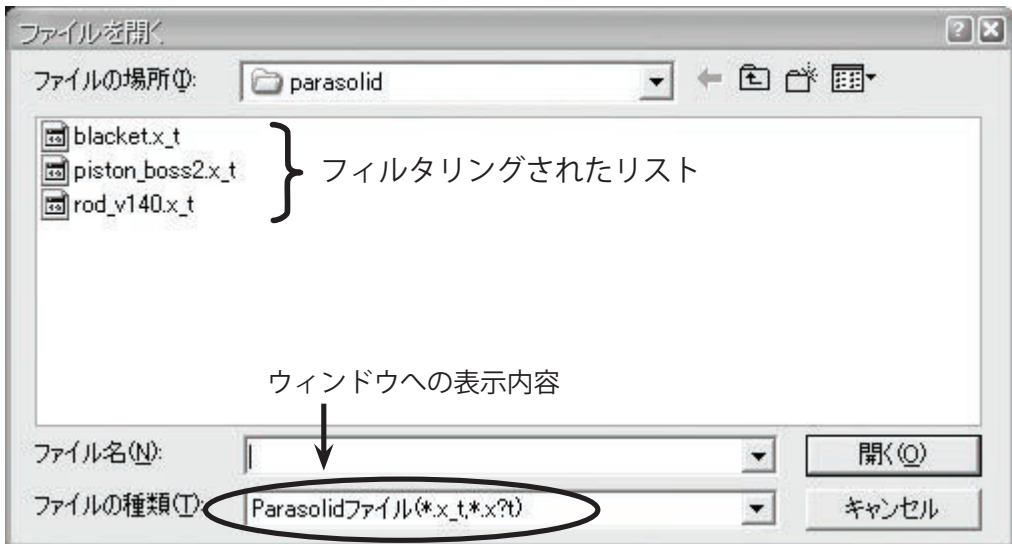
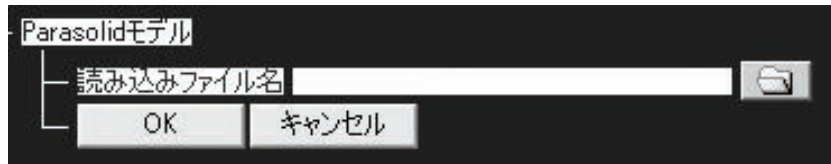
属性

属性名	内容	型	必須
CtrlID	固有の ID	STR	○
Label	ボックス説明	STR	×
Save	ファイル保存用のウィンドウを表示する場合、引数を yes として指定する	STR	×
Size	ボックスのサイズ (下記のいずれかを指定) : S: サイズ小 / M: サイズ中 (デフォルト) / L: サイズ大	STR	×
	数値: 数値によるサイズ指定 (単位は 10 ピクセル)	INT	×
Extension	ファイル選択時にフィルタリングを行う場合、その拡張子を指定。 指定方法は下記の通り: " ウィンドウへの表示内容 フィルタリングの内容* " ※複数記述する場合は "; (セミコロン) " で区切る 例) IGES ファイルの場合 "IGES ファイル (*.iges,*.igs) *.iges;*.igs " ※ Extension を省略した場合は、すべてのファイルを表示する	STR	×
Folder	フォルダ選択用のウィンドウを表示する場合、引数を yes として指定する。	STR	×

使用例

- Parasolid ファイルを選択する場合。

```
<MENU Name="Parasolidモデル" MenuID="model_para">
  <CMD Type="ok_can">
    <PROGRAM>1003</PROGRAM>
    <FILESELECT CtrlID="f_name" Label="読み込みファイル名"
      Extension="Parasolidファイル (*.x_t,*.x?t)|*.x_t;*.x?t|"/>
  </CMD>
</MENU>
```



<GET>

概要

形式	<GET Variable= ... />
包含可能要素	なし

説明

- データ保持領域の変数の値を取得します。

属性

属性名	内容	型	必須
Variable	データ保持領域の変数名	STR	○

使用例

- 固有の ID (CtrlID) local_ver に、データ保持領域の変数 global_ver の値を代入する場合。

```
<INPUT CtrlID="local_ver"><GET Variable="global_ver"/></INPUT>
```

<GUIDE>

概要

形式	<GUIDE> ... </GUIDE>
包含可能要素	ガイドを記述した html ファイル名 (パス名も含む)

説明

- ガイドを表示します。

属性

- なし

使用例

- ガイドを記入したファイル model.html を表示する場合。

```
<GUIDE>guide¥model.html</GUIDE> ← 相対パスの場合
<GUIDE>C:¥foo¥model.html</GUIDE> ← 絶対パスの場合
```

注) 相対パスの場合は NAVIAEO.exe が存在するフォルダからのパスを記入する。

<INPUT>

概要

形式	<INPUT CtrlID=... > ... </INPUT>
包含可能要素	<GET>

説明

- 固有の ID を定義し、値を代入します。
- ここで定義される変数は、<INPUT> タグを挟み込んでいる 1 階層上の <CMD> タグ内のみで有効です。

属性

属性名	内容	型	必須
CtrlID	固有の ID	STR	○
Type	ID のタイプ（下記のいずれかを指定）： string：文字列（デフォルト） integer：整数 real：実数	STR	×
Value	ID に代入する値（ID に直接値を代入する場合に使用）	STR	×

使用例

- 固有の ID（local_ver）を定義し、データ保持領域の変数 global_ver の値を代入する場合。

```
<INPUT CtrlID="local_ver"><GET Variable="global_ver"/></INPUT>
```

<ITEMCREATE>

概要

形式	<ITEMCREATE> ... </ITEMCREATE >
包含可能要素	<GET>, デフォルト値

説明

- コンボボックス <COMBO> やリストボックス <LIST> に表示するカンマ区切りデータを作成します。

属性

- なし

使用例

- <COMBO> タグ (p. 13) 及び <LIST> タグ (p. 22) を参照。

<KNOWHOW>

概要

形式	<KNOWHOW> ... </KNOWHOW>
包含可能要素	ノウハウを記述した HTML ファイル名 (絶対パスを含む)

説明

- ノウハウを表示します。

属性

- なし

使用例

- ノウハウを記入したファイル knowhow_1.html を表示する場合。

```
<KNOWHOW>C:¥foo¥knowhow_1.html</KNOWHOW> ← 絶対パスのみ可
```

<LIST>

概要

形式	<LIST CtrlID= ... > ... </LIST>
包含可能要素	<GET>, <ITEMCREATE>, デフォルト値

説明

- 選択可能なパラメータをリスト形式で表示します。固有の ID を定義し、選択された項目の番号をこの ID に保存します。(デフォルト値にはカンマ区切りリストの何番目かを指定します。)

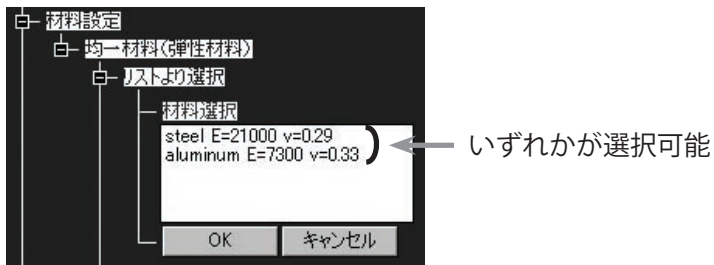
属性

属性名	内容	型	必須
CtrlID	固有の ID	STR	○
Label	ボックス説明	STR	×
Size	ボックスのサイズ (下記のいずれかを指定) : S: サイズ小 / M: サイズ中 (デフォルト) / L: サイズ大	STR	×
	数値 : 数値によるサイズ指定 (単位は 10 ピクセル)	INT	×
Type	リスト項目の選択種別 (下記のいずれかを指定) : single : 複数選択不可 (デフォルト) multi : 複数選択可 (※複数選択時はカンマ区切りで値が返ります)	STR	×

使用例

- 材料ライブラリをリスト表示する。どれか1つのみ選択可能。

```
<MENU Name=" リストより選択 " MenuID="mat_st_list">
<PROGRAM Lang="Perl" PrgFile="mat_list_cre.pl"
CallFunc="main_func"/>
<CMD Type="ok_can">
  <SET Variable="sel_mat" Type="string" IDValue="sel_mat"/>
  <PROGRAM Lang="Perl" PrgFile="comformp.pl" CallFunc="list_mat"/>
  <LIST CtrlID="sel_mat" Label=" 材料選択 ">
    <ITEMCREATE>
      <GET Variable="mat_list"/>
    </ITEMCREATE>
  </LIST>
</CMD>
</MENU>
```



<MENU>

概要

形式	<MENU Name= ... > ... </MENU>
包含可能要素	<MENU>, <SET>, <CMD>, <PROGRAM> [*] , <GUIDE>, <KNOWHOW> (※ : Perl 使用時のみ)

説明

- メニューツリーの各項目を表示します。

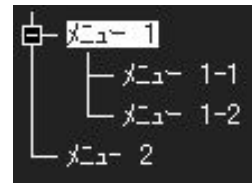
属性

属性名	内容	型	必須
MenuID	グレーアウト処理用のメニュー識別番号	STR	×
Name	ツリーメニューの表示名.	STR	○
Tip	ツールチップの内容.	STR	×

使用例

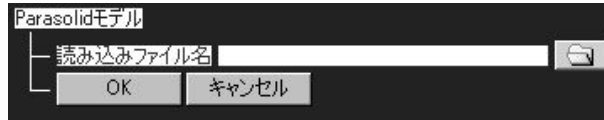
- 2 階層のツリーメニューを作成する場合。

```
<MENU Name="メニュー 1">
  <MENU Name="メニュー 1-1" MenuID=" me1-1" >
    (メニュー 1-1 の処理内容を記述)
  </MENU>
  <MENU Name="メニュー 1-2" MenuID=" me1-2" >
    (メニュー 1-2 の処理内容を記述)
  </MENU>
</MENU>
<MENU Name="メニュー 2">
  (メニュー 2 の処理内容を記述)
</MENU>
```



- Parasolid ファイルを読み込むメニューの場合。

```
<MENU Name="Parasolid モデル " Tip=" ファイルを読み込みます。 "  
MenuID="model_para">  
  <CMD Type="ok_can">  
    <PROGRAM>1003</PROGRAM>  
    <FILESELECT CtrlID="f_name" Label=" 読み込みファイル名 "  
      xtension="Parasolid ファイル (*.x_t,*.x?t)|*.x_t;*.x?t|"/>  
  </CMD>  
</MENU>
```



<NaviaXml>

概要

形式	<NaviaXml> ... </NaviaXml>
包含可能要素	<TREE>

説明

- 解析メニューツリーのルートタグを定義します。
- ヘッダ部のタグ <?xml ... ?> とともに必ず記述します。

属性

- なし

使用例

- NAVIAEO 用の xml であることを宣言。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<NaviaXml>
  <TREE ...

  (NAVIAEO の xml を記述)

  </TREE>
</NaviaXml>
```

注意事項

- NAVIA ツリー xml ファイル内でのみ使用可能。

<PROGRAM>

概要

形式	<PROGRAM> ... </PROGRAM>
包含可能要素	コマンドの識別番号 (Perl プログラムの場合はなし)

説明

- コマンド^{※注}、又は Perl プログラムを実行します。

属性

- コマンドを実行する場合：なし
- Perl プログラムを実行する場合：

属性名	内容	型	必須
Lang	プログラムの言語 "Perl" を指定	STR	○
PrgFile	プログラムのファイル名	STR	○
CallFunc	ファイル内のメイン関数名	STR	○

使用例

- コマンド (EXEL) を実行する場合。

```
<PROGRAM>EXEL</PROGRAM>
```

- Perl プログラムファイル comformp.pl 中の関数 list_mat を実行する場合。

```
<PROGRAM Lang="Perl" PrgFile="comformp.pl" CallFunc="list_mat"/>
```

注意事項

- コマンドの場合は <CMD> タグ内でのみ使用可能 (複数不可)。Perl プログラムの場合は <MENU> 及び <CMD> タグ内で使用可能 (複数可)。
- 複数の <PROGRAM> タグを連続して使用してカスタマイズしている場合、エラーが発生した <PROGRAM> タグより後ろの行に記述されている <PROGRAM> タグは無視されます。

※注) 詳細は 6. NAVIA MANAGER - コマンド登録機能を参照ください。

<ROOT>

概要

形式	<ROOT> ... </ROOT>
包含可能要素	<MENU>

説明

- ツリーセレクトツリーのルートタグを定義します。
- ヘッダ部のタグ <?xml ... ?> とともに必ず記述します。

属性

属性名	内容	型	必須
Name	ルートの表示名	STR	○

使用例

- <TREETYPE> タグを参照。

注意事項

- ツリーセレクト xml ファイル (AnaType_(I/F の固有名称).xml) 内でのみ使用可能。

<SET>

概要

形式	<SET Type= ... Variable= ... />
包含可能要素	なし

説明

- データ保持領域に変数を定義し、値を入力します。

属性

属性名	内容	型	必須
IDValue	変数の値 (CtrlID による入力の場合)	STR	△
Type	変数の型 整数の場合：integer 実数の場合：real 文字列の場合：string グレースアウトの場合：gray を指定	STR	○
Update	上書きの可・不可 可の場合：yes (デフォルト) 不可の場合：no	STR	×
Value	変数の値 (直接入力の場合)	STR	△
Variable	変数名	STR	○

注) ID Value と Value のいずれかを指定する必要があります。

使用例

- データ保持領域に変数 mat_name を定義し、文字列 “steel” を入力する場合。

```
<SET Variable="mat_name" Type="string" Value="steel"/>
```

- データ保持領域に変数 mat_name を定義し、CtrlID= “matID” で参照される値を入力する場合。

```
<SET Variable="mat_name" Type="string" IDValue="matID"/>
```

<STATIC>

概要

形式	<STATIC CtrlID= ... > ... </STATIC>
包含可能要素	<GET>, デフォルト値

説明

- パラメータ入力用のボックスを表示します。固有の ID を定義し、ボックスに入力された値をこの ID に保存します。ユーザーによる修正は不可です。

属性

属性名	内容	型	必須
CtrlID	固有の ID	STR	×
Label	ボックス説明	STR	×

使用例

- ポアソン比をデフォルト値 0.3 に設定して画面へ表示する場合。材料名とヤング率はユーザー側で入力できますが、ポアソン比は 0.3 に固定されます。

```
<EDIT CtrlID="mat_name" Label="材料名" />
<EDIT CtrlID="young" Label="ヤング率" />
<STATIC CtrlID="poisson" Label="ポアソン比">0.3</STATIC>
```

材料名	<input type="text"/>
ヤング率	<input type="text"/>
ポアソン比	0.3

<TREE>

概要

形式	<TREE> ... </TREE>
包含可能要素	<MENU>

説明

- メニューツリーを定義します。

属性

属性名	内容	型	必須
TreeID	ツリーの識別名 (<TREETYPE> 中の TreeID と同じ値にする必要がある)	STR	○
Software	NAVIAEO が扱うソフトウェア名	STR	×
Version	NAVIAEO が扱うソフトウェアのバージョン	STR	×

使用例

- 2つのメニューツリーを定義する場合。

```
<TREE TreeID="stress" Software="000" Version="000">
```

```
(stressに関するメニューツリーの記述)
```

```
</TREE>
```

```
<TREE TreeID="vibr" Software="000" Version="000">
```

```
(vibrに関するメニューツリーの記述)
```

```
</TREE>
```

<TREETYPE>

概要

形式	<TREETYPE> ... </TREETYPE>
包含可能要素	なし

説明

- NAVIA ツリーのパラメータを定義します。

属性

属性名	内容	型	必須
TreelD	NAVIA ツリーの識別名 (<TREE> タグ中の TreelD と同じ値にする必要がある)	STR	○
TreeFile	NAVIA ツリーを記述している xml 名	STR	○
Type	NAVIA ツリーを実行するボタンのタイプ (下記のいずれかを指定) : ok, ok_can, conf_ok_can	STR	○

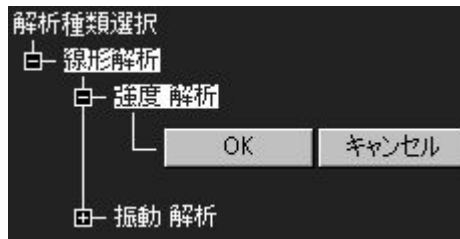
- ボタンのタイプについて

タイプ	内容
ok	OK ボタンを表示する。OK ボタンを押すとプログラムを実行し、メニューを閉じる
ok_can	OK ボタンとキャンセルボタンを表示する。OK ボタンを押すとプログラムを実行し、メニューを閉じる。キャンセルボタンを押すとプログラムを実行せずにメニューを閉じる
conf_ok_can	確認ボタン、OK ボタン及びキャンセルボタンを表示する。初期状態では確認ボタンとキャンセルボタンが有効である。確認ボタンを押した後、OK ボタンが有効になる。OK ボタンを押すとプログラムを実行し、メニューを閉じる。キャンセルボタンを押すとプログラムを実行せずにメニューを閉じる

使用例

- 2つの解析タイプ（強度解析と振動解析）を定義する場合。
- NAVIA ツリーは myFile.xml に記述されています。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<Root Name=" 解析種類選択 ">
  <MENU Name=" 線形解析 ">
    <MENU Name=" 強度解析 " Tip=" 線形強度解析を行います。">
      <TREETYPE TreeID="stress" TreeFile="myFile" Type="ok_can"/>
    </MENU>
    <MENU Name=" 振動解析 " Tip=" 振動解析を行います。">
      <TREETYPE TreeID="vibr" TreeFile="myFile" Type="ok_can"/>
    </MENU>
  </MENU>
</Root>
```



注意事項

- ツリーセレクト xml ファイル（AnaType_(I/F)の固有名称).xml) 内でのみ使用可能。

<!-->

概要

形式	<!-- ... -->
包含可能要素	なし

説明

- コメント行を作成します。ツリーには反映されません。

属性

なし

使用例

- コメントを作成する場合。

```
<!-- これは コメントです。 -->
```

<?xml>

概要

形式	<?xml ... ?>
包含可能要素	<NaviaXml>

説明

- このファイルが xml であることを宣言します。
- 必ずファイルの先頭行に記述します。

属性

属性名	内容	型	必須
version	xml の書式のバージョン	STR	○
encoding	エンコーディングの種類	STR	○

使用例

- xml を宣言します。
- 特に指定がない限り、必ず下記のように記述してください。

```
<?xml version="1.0" encoding="Shift-JIS" ?>
```

- <NaviaXml> (p. 26) タグ参照。

4. Perl によるカスタマイズ

Perl を使うことで、xml タグやコマンドでは対応できないような処理を行うことが可能です。

例) 入力方法の変更、文字列処理など

- Perl と xml タグ間のデータの受け渡しには、「データ保持領域」に宣言した変数を使用します。
- この章では、Perl を使ってカスタマイズする場合の概要及びカスタマイズ事例を説明します。事例としては、下記を取り上げます。

◇ 負荷条件の入力方法を変更する。

※本書では Perl の文法自体については説明いたしません。Perl に関する専門書をご覧ください。

カスタマイズの概要

Perl ファイルの保管場所

- Perl ファイルは下記のフォルダへ保存してください。

保存先フォルダ：<NAVIAEO インストールフォルダ>%bin%perl
ファイルの拡張子は“.pl”です。

NAVIAEO カスタマイズ機能

NAVIAEO に対して Perl で利用できる主な機能は下記の通りです。

Perl ファイルサンプル例

```
sub main_func{
#<GETVAL>$MyFile ..... 1
#<UPDVAL_STR>$ErrMsg ..... 4
$ErrMsg="";
#<UPDVAL_STR>$CmdStr ..... 2
#<CMD>$CmdStr ..... 3

if($MyFile ne ""){
$CmdStr="Save,$MyFile";}
else{$ErrMsg="ファイルがありません。";}
}
```

1 NAVIAEO データ保持領域内のデータを取得する。

サンプル例の通り「#<GETVAL> + 変数名 (NAVIAEO と共通の変数名)」と記述することにより Perl 内に変数値を取得することができます。

※ NAVIAEO に初期値として用意されている変数もあります (「6. その他」を参照)。

2 NAVIAEO データ保持領域内にデータを保持する。

サンプル例の通り「#<UPDVAL_STR> + 変数名 (NAVIAEO に同じ変数名で保存)」と記述することにより NAVIAEO に値が保存されます。なお、設定するものとして下記の種類があります。

#<UPDVAL_STR>	NAVIAEO に値がストリング型で上書き保存
#<SETVAL_STR>	NAVIAEO に値がストリング型で保存 (上書き不可)

※保存する変数が Perl 内で初期化されていないと、エラーになる可能性があります。

3 エラー時にエラーメッセージを NAVIAEO で出力する。

「\$ErrMsg」は予約変数です。値が入った「\$ErrMsg」を NAVIAEO が受け取るとエラーメッセージとして表示します。

注意事項：

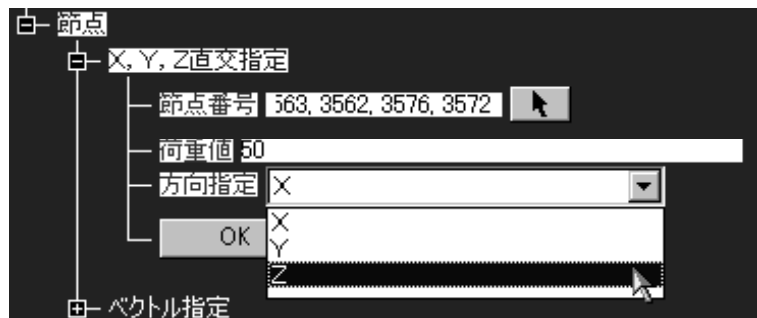
- "\$ErrMsg" でエラーが発生した場合、他の変数は保存されません。
- 複数の<PROGRAM> タグが別のタグ内に記述されている場合、<PROGRAM> タグは降順に(上から下へ)実行されます。
- もし<PROGRAM> タグのどれか一つでエラーが発生した場合、エラーが発生した<PROGRAM> タグ及びそれ以降の<PROGRAM> タグは実行されません。

カスタマイズ事例

負荷条件の入力方法を変更する

汎用コマンド 1511 では負荷荷重の成分値を直接入力する必要があります。もし直交座標系の X, Y, Z 方向のみで荷重方向を指定したい場合は、<COMBO> タグを使い xml ファイルを下記のように記述します。

```
<MENU Name="X, Y, Z 直交指定 " MenuID="force_node_xyz">
  <CMD Type="ok_can">
    3 { <SET Variable="xyz_val" Type="string" IDValue="xyz_val"/>
      <SET Variable="load_val" Type="string" IDValue="load_val"/>
    4 . . . . . <PROGRAM Lang="Perl" PrgFile="force_node.pl" CallFunc="main_func"/>
      <EDIT CtrlID="load_val" Label=" 荷重値 "/> . . . . . 2
      <COMBO CtrlID="xyz_val" Label=" 方向指定 ">
        <ITEMCREATE>
          X, Y, Z
        </ITEMCREATE>
      </COMBO>
    } 1
  </CMD>
</MENU>
```

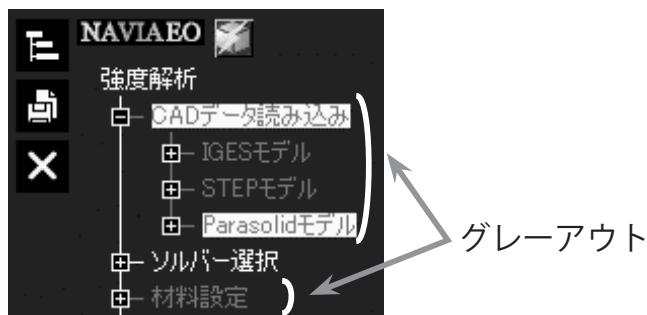


- 1** 負荷方向についてはコンボボックスで X,Y,Z のいずれかを選択し、変数 (CtrlID) xyz_val に代入します。
- 2** 荷重値は直接入力に変数 (CtrlID) load_val に代入します。
- 3** <SET> タグでデータ保持領域の変数 xyz_val、node_ID、および load_val を定義し、**1** ~ **3** で入力された値をこれらの変数に代入します。
- 4** Perl プログラム、force_node.pl 内の関数 main_func を実行します。force_node.pl は次のようになっています。

5. グレーアウトの設定

グレーアウトを設定することにより、NAVIAEO のメニューツリーを選択可／不可にすることが可能です。

グレーアウトを設定することで、作業の流れに従ってユーザーが操作可能なメニューを指定することができ、作業をスムーズに行うことが可能になります。



グレーアウト設定ファイルの保管場所

グレーアウト設定ファイルは下記のフォルダへ保存してください。

<NAVIAEO インストールフォルダ>¥xml

ファイルの拡張子: .gry

グレーアウトの設定方法

- グレーアウト処理では xml 内の MenuID を判断材料に利用します。<MENU> タグの下位に <CMD> タグがある場合、OK ボタンが押されるとその MenuID が“済 (1)”として記憶されます。
- グレーアウトを制御したい MenuID を左辺に、条件を右辺に書き、その条件式が“1”を満たす場合、グレーアウトが解除されます。
- 但し、左辺として記述の無い MenuID は無条件で選択可能（グレーアウトしない）状態になります。尚、条件式には AND、OR、NOT が使用可能です。

グレーアウトの設定例

xml ファイル (抜粋)

```
<MENU MenuID="id1">
  <MENU MenuID="id1_1">
    <CMD Type="ok"/>
  </MENU>
  <MENU MenuID="id1_2">
    <MENU MenuID="id1_2_1">
      <CMD Type="ok"/>
    </MENU>
  </MENU>
</MENU>
<MENU MenuID="id2">
  <MENU MenuID="id2_1">
    <CMD Type="ok"/>
  </MENU>
  <MENU MenuID="id2_2">
    <CMD Type="ok"/>
  </MENU>
  <MENU MenuID="id2_3">
    <CMD Type="ok"/>
  </MENU>
</MENU>
```

グレーアウト設定ファイル

```
[grayout]
id1   = NOT(id1_2_1) ..... 1
id1_2 = id1_1 ..... 2
id2   = AND(id1_1,id1_2).... 3
id2_3 = OR(id2_1,id2_3).... 4
```

- 1** “id1_2_1” の OK ボタンが押されると “id1” がグレーアウトする。
- 2** “id1_1” の OK ボタンが押されると “id1_2” のグレーアウトが解除される。
- 3** “id1_1”、“id1_2” の両 OK ボタンが押されると “id2” のグレーアウトが解除される。
- 4** “id2_1” または “id2_3” の OK ボタンが押されると “id2_3” のグレーアウトが解除される。

6. NAVIA MANAGER

コマンド登録機能

様々なソフトウェア (実行ファイル) をコマンドとして登録することにより NAVIA MANAGER から起動させることができます。(最大5個まで)

1. Setting.xml には下の通りに記述します

【SET タグ】

- [Manager 属性] に動作させたいコマンド名を記述します
- [Value 属性] に動作させたいソフトウェアのフルパスを記述します

```
<SET Manager="IE" Type="string" Value="C:\Internet Explorer\IEXPLORE.EXE"/>
<SET Manager="EXCEL" Type="string" Value="C:\Program Files\Office\EXCEL.EXE"/>
```

Setting.xml の記入例

2. NAVIA ツリー XML には下の通り記述します

【PROGRAM タグ】

- Setting.xml で作成したコマンドは PROGRAM タグで挟む事で使用できます。(引数も EDIT タグ等を利用して付与することができます)

```
<?xml version="1.0" encoding="Shift-JIS" ?>
<NaviaXml>
  <TREE TreeID="Manager" Software="Manager">
    <MENU Name="IE" Tip="Web ブラウザの起動" MenuID="soft1">
      <CMD Type="ok_can">
        <PROGRAM>IE</PROGRAM>
        <EDIT CtrlID="IE"></EDIT>
      </CMD>
    </MENU>
    . . . <以下省略> . . .
```

NAVIA ツリー XML の記入例

7. その他

NAVIAEO で使用する予約変数について説明します。

NAVIAEO の予約変数

予約変数とは

予約変数とは NAVIAEO が必要に応じて参照する変数のことです。NAVIAEO にはいくつかの予約変数があり、NAVIAEO の起動時などにデータが設定されます。

予約変数の一覧

NAVIAEO には下記のような予約変数があります。

変数名	内容
PERL_PATH	Perl ファイルを保存するフォルダのフルパス名 (NAVIAEO インストールフォルダ) %bin%perl
SYSTEM_PATH	NAVIAEO.exe が存在するフォルダのフルパス名
WORK_PATH	NAVIAEO が一時的に使用する作業フォルダのフルパス名
TREEID	ユーザが NAVIAEO で選択したツリーの種類に対応する <TREETYPE> タグ中の TreeID のデータ
TREENAME	ユーザが NAVIAEO で選択したツリーの種類に対応する <TREETYPE> タグの 1 つ上の階層にある <MENU> タグ中の Name のデータ
NAV_solver	NAVIAEO 起動時に選択したサブモジュール

これらの予約変数はデータ保持領域に定義されますので、xml の <GET> タグや Perl プログラムを使ってユーザーが取得することも可能ですが、xml の <SET> タグや Perl プログラムで上書きしないように注意してください。

NAVIAEO の予約コマンド

予約コマンドとは

予約コマンドとは NAVIAEO の操作に必要なコマンドとして用意されているものです。NAVIAEO にはいくつかの予約コマンドがありますので必要に応じてご利用下さい。

予約コマンドの一覧

NAVIAEO には下記のような予約コマンドがあります。

コマンド名	内容
SAVE	ファイルパスを指定することで、NAVIAEO の状態を保存するコマンドです (汎用版 HyperMesh では HyperMesh の保存も行われます)
LOAD	ファイルパスを指定することで、拡張子 (.NAVI) のファイルを読み込み、以前に保存した NAVIAEO の状態に戻します
OPEN	ファイルパス (※ 1) を指定することで、一般的なソフト (表計算ソフト、Web ブラウザ等) を起動させることができます ※ 1 実行可能形式のファイル (exe、bat 等)、もしくはプログラムに関連付けされた拡張子のファイルを指定可能です。 なお、関連付けはご利用される環境によって異なります。Windows のフォルダオプション等を参照して下さい。

これらの予約コマンドは <PROGRAM> タグを利用して使用することができます。詳しくは <PROGRAM> タグ (p.27) を参照してください。

Notes

Installation and Operation Guide NAVIAEO v.1.5 インストール&操作・機能ガイド

平成 21 年 01 月 27 日 第 2 版発行

発行 インテグラル・テクノロジー株式会社
〒 550-0011
大阪府大阪市西区阿波座 2-1-1
大阪本町西第一ビルディング 11F

©2007 インテグラル・テクノロジー株式会社

本書中に登場する会社名や商標名は一般に各社の商標または登録商標です。

NAVIAEO VER.1.5

インストール&機能ガイド

Installation and **Operation** Guide

