

# CONTENTS

---

ビジュアルヘルプ - Duplicate.....	2
Duplicate コマンドのヘルプ.....	2

# ビジュアルヘルプ – Duplicate

## Duplicate コマンドのヘルプ

Duplicate [*flags*][*type flags*] *srcWaveName*, *destWaveName* [, *destWaveName*]...

Duplicate コマンドは、新しいウェーブを作成します。

その名前は *destWaveName* で指定され、その内容、データ形式、スケーリングは *srcWaveName* と同じです。

### パラメーター

*srcWaveName* は既存のウェーブの名前でなければなりません。

*destWaveNames* は、既存のウェーブを上書きする /O フラグを使わない限り、現在使用していないウェーブ名である必要があります。

### フラグ

/FREE[=*nm*]

フリーウェーブを作成します。

関数内でのみ使用可能で、単純な名前またはウェーブ参照構造フィールドが指定されている場合のみ許可されます。

*nm* が指定され、ゼロでない場合、*waveName* はフリーウェーブの名前として使われ、デフォルト名「\_free\_」を上書きします。

フリーウェーブの名前を指定する機能は、デバッグ支援として Igor Pro 9.0 で追加されました。

/O

*destWaveName* と同じ名前の既存のウェーブを上書きします。

/R=(*startX*,*endX*)

ソースウェーブの X の範囲を指定し、その範囲から目標のウェーブが作成されます。

/R=(*startX*,*endX*)  
(*startY*,*endY*)

X と Y の両方の範囲を指定します。

さらに、次元を同様に作成します。

/R=[*startP*,*endP*]

ソースウェーブの行の範囲を指定し、そこから目標のウェーブが作成されます。

その他の次元はスケーリングされた次元範囲と同じように構築されます。

/RMD=[*firstRow*,*lastRow*]  
[*firstColumn*,*lastColumn*]  
[*firstLayer*,*lastLayer*]  
[*firstChunk*,*lastChunk*]

コマンドを適用するソースウェーブの連続したデータの範囲を指定します。

このフラグは Igor Pro 7.0 で追加されました。

対応する括弧を省略することで、すべての高次元を含めることができます。

例えば、

/RMD=[*firstRow*,*lastRow*]

は、すべての使用可能な列、レイヤー、チャンクを含みます。

空の [] を使うと、指定された次元をすべて含めることができます。

例えば、

```
/RMD=[] [firstColumn, lastColumn]
```

は、「A 列から B 列までのすべての行」を意味します。

\* を使って、任意の次元の終端を指定することができます。

例えば、

```
/RMD=[firstRow, *]
```

は、「最初の行から最後の行まで」を意味します。

## 形式フラグ (関数でのみ使われる)

ユーザー定義関数で使われる場合、Duplicate は、/B、/C、/D、/I、/S、/U、/W、/T、/DF、/WAVE フラグも使うことができます。

これらは、Duplicate コマンドの結果には影響しません。

これらのフラグは、実行時にどのようなウェーブが予想されるかを識別するためにのみ使われます。

この情報は、関数の後で、複製したウェーブを目標として使うウェーブ代入文を作るときに使われます。

```
Function DupIt (wv)
    Wave/C wv // 複素数ウェーブ
    Duplicate/O/C wv, dupWv // dupWv が複素数であることを Igor に指示
    dupWv[0]=cmplx(5.0, 1.0) // エラーなし。dupWv は既知の複素数ウェーブであるため
    ...
```

Duplicate に /C フラグが指定されていなかった場合、Igor は cmplx 関数の結果への dupWv の代入文をコンパイルしようとする、「function not available for this number type (この数値型では関数が使用できません)」というエラーメッセージを表示します。

これらの形式フラグは、同じ名前の別のウェーブ参照変数と一致させる必要がある場合、または、ウェーブ代入用にコンパイルする式の種類を識別する必要がある場合を除いて、使う必要はありません。

形式フラグの完全なリストと詳細な説明は、ヘルプ WAVE Reference Types と WAVE Reference Type Flags を参照してください

## 詳細

/R が省略された場合、ウェーブ全体が複製されます。

/R とともに使われる範囲指定において、末尾に \* を付けると、末尾まで複製することを意味します。

また、単に末尾の指定を省略することもできます。

指定した次元をすべて含めるには、

```
/R=[]
```

を使います。

高次元を省略すると、それらの次元はすべて複製されます。

つまり、2D ウェーブの

```
/R=[1, 5]
```

は、

```
/R=[1, 5] []
```

と同等です。

ソースのウェーブがロックされていた場合でも、目標のウェーブは常にロックが解除されます。

### 警告

ユーザー定義関数のループなど、一部の状況においては、Duplicate が予期しない動作を示す場合があります。

ユーザー定義関数内で、

```
Duplicate/O srcWave, DestWaveName
```

を使うとき、これはコンパイル時に DestWaveName というローカル WAVE 変数を作成します。

実行時に WAVE 変数が NULL の場合、現在のデータフォルダーに同じ名前のウェーブが作成されます。

しかし、ループ内であれば WAVE 変数が NULL ではないため、参照されたウェーブはその場所に関わらず上書きされます。

現在のデータフォルダーにウェーブを作成（または上書き）したい場合は、以下の2つの方法のうちの1つを使用してください。

```
Duplicate/O srcWave, $"DestWaveName"
```

```
WAVE DestWaveName
```

**// 目標のウェーブを参照する必要がある場合のみ**

または

```
Duplicate/O srcWave, DestWaveName
```

```
// DestWaveName の使用が完了したら
```

```
WAVE DestWaveName="$"
```

### 参照

Rename, Concatenate, SplitWave