



# クライアントサイド でのデータ認証の 使い方

---

© 2002-2007 Unify Corporation All rights reserved. Sacramento California, USA

No part of this tutorial may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written consent of Unify Corporation.

Unify Corporation makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Unify Corporation reserves the right to revise this document and to make changes from time to time in its content without being obligated to notify any person of such revisions or changes.

The Software described in this document is furnished under a Software License Agreement. The Software may be used or copied only in accordance with the terms of the license agreement. It is against the law to copy the Software on tape, disk, or any other medium for any purpose other than that described in the license agreement.

The Unify Corporation Documentation Group values and appreciates any comments you may have concerning our documents. Please address comments to:

doc@unify.com

1-800-24 UNIFY or 1-800-GO-UNIFY;(916) 928-6400  
FAX (916) 928-6401

UNIFY and DataServer are registered trademarks of Unify Corporation. Unify NXJ is a trademark of Unify Corporation. Java and J2EE are registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. JReport is a trademark of Jinfonet Corporation. IBM, Lotus, Lotus Notes, Cloudscape, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. CAS AHL Technology and ecKnowledge are registered trademarks of CAS AHL Technology, Inc. in the U.S. and other countries. All other products or services mentioned herein may be registered trademarks, trademarks, or service marks of their respective manufacturers, companies, or organizations.

Name: Working with Client-Side Validation

Release: Unify NXJ 11.7/Composer

Last Revision: January 23, 2009 2:58 pm

---

# クライアントサイドでの データ認証の使い方

このドキュメントでは、クライアントサイドでのデータ認証について説明します。クライアントサイドでのデータ認証は、1) 必須データコントロールに対してデータが入力されたかどうか、2) フィールドコントロールに対して正しいデータが入力されたか、をチェックするために使用します。この処理はクライアント側で、データがアプリケーションサーバに送られる前に実行されます。

## 処理の概要

まず、どのコマンド（Add Update、Next Form など）がデータ認証を呼び出すかを指定します。ユーザがこれらのコマンドを実行した場合、NXJ は、まだサーバに送られていない必須フィールドコントロールをチェックし、そのフィールドコントロールに割り当てられた JavaScript で書かれた認証ルールを実行します。必須フィールドコントロールにデータが入力されていない場合、NXJ はエラーメッセージを表示し、フィールドコントロールの境界線を強調表示します。また、JavaScript で書かれた認証ルールに合致しない場合には、NXJ は認証に失敗した最初フィールドコントロールにフォーカスを移し、境界線を強調表示します。そのフィールドコントロールがタブに存在する場合、NXJ はそのフィールド含むタブを表示します。

## プロパティ

以下の表に各プロパティおよびそのプロパティを使用するクラスの一覧を示します。

プロパティ	使用方法	関係するクラス
Component > Includes	NXJ がデータ認証のために使用する JavaScript ソースファイルを指定します。ファイルを指定すると、各フィールドコントロールの Validation Rules プロパティに、実行させる JavaScript 関数を指定することができるようになります。	Form BPM Activity Form Checkbox Date Picker Dropdown List JavaScript Field Listbox Radio Group Text Area Text Field Box Data View
Interactive > Required for Command	フォームに割り当てられた認証コマンドを実行する際、このフィールドコントロールにデータが入力されていなければならないことを示します。例えば、Address フォームで Street フィールドを必須とし、Add Update コマンドが認証コマンドとして指定されている場合、ユーザが Street データを指定せずに、Address を追加、更新しようとする NXJ はエラーメッセージを表示します。	Date Picker Dropdown List JavaScript Field Listbox Text Area Text Field
Interactive > Validation Commands	データ認証を実行するコマンド (Add Update, Exit, Previous Form 等) を指定します。	Form BPM Activity Form
Interactive > Validation Rules	NXJ がフィールドコントロールのデータ認証に使用する JavaScript 関数を指定します。この関数は、フィールドコントロールの Include プロパティで指定された JavaScript ファイル中で定義されていなければなりません。	Checkbox Date Picker Dropdown List JavaScript Field Listbox Radio Group Text Area Text Field Data View
Visual > Error Border Color	データ認証に失敗したフィールドコントロールの境界に使用する色を指定します。デフォルトは赤です。	Form BPM Activity Form

## フィールドコントロールを必須項目にする

ユーザが特定のコマンドを実行する際に必須となるフィールドコントロールを指定することができます。この種類の認証を使用するには、まず、どのコマンド (Add Update, Next Form 等) を使用するかを、認証が必要なフィールドコントロールを持つフォームで指定します。Interactive グループの Validate Commands プロパティにコマンドを指定できます。

---

データ認証を実行するコマンドを指定した後、必須となるフィールドコントロールを指定します。フィールドコントロールを選択し、Interactive グループの Required for Command チェックボックスをチェックして下さい。

## フィールドコントロールに認証ルールを設定

以下の手順でフィールドコントロールに認証ルールを割り当てることが出来ます。

- 認証時に実行されるコマンドを指定
- JavaScript ソースファイルを作成
- JavaScript ソースファイルの指定
- 実行する JavaScript 関数を指定

### タスク 1: 認証時に実行されるコマンドを指定

どのコマンド (Add Update, Next Form 等) を使用するかを、認証が必要なフィールドコントロールを持つフォームで指定します。Interactive グループの Validate Commands プロパティにコマンドを指定できます。

### タスク 2: JavaScript ソースファイルを作成

プロジェクトの Static Content フォルダに JavaScript ファイルを作成します。このファイルには、NXJ がデータ認証に使用する関数を定義しておきます。

各関数には、フィールドコントロールを引数として指定することができます。関数は、認証に失敗した場合には false を返し、成功した場合には true を返す必要があります。以下に例を示します。

```
function myValidator(myControl)
{
    if (myControl.value == "0")
    {
        alert("Zero is not allowed.");
        return false;
    }
    return true;
}
```

---

## タスク 3: JavaScript ソースファイルの指定

データ認証関数を定義した後、認証が必要なフィールドコントロールに JavaScript ソースファイルを指定します。フィールドコントロール、またはそのコントロールを含むフォームを選択し、Component グループの Includes プロパティに、JavaScript ファイル名を指定します。

## タスク 4: 実行する JavaScript 関数を指定

JavaScript ソースファイルを指定した後、Interactive グループの Validation Rules プロパティに、関数名を指定します。複数の関数を指定する場合は、各関数をセミコロン (;) で区切って指定します。関数名のみを指定し、引数またはパラメータは指定しないようにして下さい。以下に例を示します。

```
myValidator;ZIPValidator;phoneValidator
```

## ‘validation\_error’ フィールドの使用

NXJ では、認証エラーメッセージをポップアップの代わりにフィールドに表示させることができます。

必須コントロールの認証エラーを表示するには、エラーメッセージを表示するテキストフィールドまたはテキストエリアを validation\_error という名称でフォームに追加します。

必須でないコントロールの認証エラーを表示するには、エラーメッセージを表示するテキストフィールドまたはテキストエリアをフォームに追加し、JavaScript ソースファイルの作成セクションで述べられているようにファンクションを追加します。このファンクションでは、テキストフィールドまたはテキストエリアの value 属性に、表示するエラーメッセージを設定します。

```
function myValidator(myControl)
{
  if (myControl.value == "0")
  {
    vfield =
    document.forms["NXJForm"].elements["validation_error."].value =
    "Zero is not allowed.";
    return false;
  }
  return true;
}
```