

# Utiliser le DynamicPopulate avec un contrôle Utilisateur et du JavaScript

par Christian Wenz Matt-K (Traducteur) ([Page d'accueil](#))


Date de publication : 04 mai 2009

Dernière mise à jour :

Le contrôle DynamicPopulate de l'ASP.NET AJAX Control Toolkit fait appel à un web service (ou un méthode de la page) et remplit le contrôle ciblé de la page avec les valeurs récupérées, sans rafraîchir la page. Il est aussi possible de lancer le remplissage automatique côté client en utilisant du JavaScript. Cependant nous devons porter une attention particulière lorsque l'extender est au sein d'un contrôle Utilisateur.

Traduction.....	3
Présentation.....	3
Étapes.....	3

## Traduction

Cet article est la traduction la plus fidèle possible de l'article original de Christian Wenz:  [Using DynamicPopulate with a User Control And JavaScript](#)

## Présentation

Le contrôle **DynamicPopulate** de l'ASP.NET AJAX Control Toolkit fait appel à un web service (ou un méthode de la page) et remplit le contrôle ciblé de la page avec les valeurs récupérées, sans rafraîchir la page. Il est aussi possible de lancer le remplissage automatique côté client en utilisant du JavaScript. Cependant nous devons porter une attention particulière lorsque l'extender est au sein d'un contrôle Utilisateur.

## Étapes

Tout d'abord, nous avons besoin d'un web service ASP.NET qui implémente la méthode qui sera appelée par le **DynamicPopulateExtender**. Le web service implémente la méthode getDate() qui requiert un argument de type string, nommée contextKey, puisque le **DynamicPopulate** envoie une information de contexte à chaque appel du service web. Voici le code (dans le fichier **DynamicPopulate.cs.asmx** qui récupère la date dans un de ces trois formats:

```
<%@ WebService Language="C#" Class="DynamicPopulate" %>
using System; using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Web.Script.Services;

[ScriptService] public class DynamicPopulate : System.Web.Services.WebService {
    [WebMethod]
    public string getDate(string contextKey) {
        string myDate = "";
        switch (contextKey) {
            case "format1":
                myDate = String.Format("{0:MM}-{0:dd}-{0:yyyy}", DateTime.Now);
                break;
            case "format2":
                myDate = String.Format("{0:dd}.{0:MM}.{0:yyyy}", DateTime.Now);
                break;
            case "format3":
                myDate = String.Format("{0:yyyy}/{0:MM}/{0:dd}", DateTime.Now);
                break;
        }
        return myDate;
    }
}
```

Ensuite, créons un nouveau contrôle utilisateur (fichier .ascx), désigné par la déclaration suivante à la première ligne:

```
<%@ Control Language="C#" ClassName="DynamicPopulate2" %>
```

Ajoutons un <label> qui sera utilisé pour afficher les données en provenance du serveur.

```
<label id="myDate" runat="server" />
```

Dans notre fichier du contrôle utilisateur, nous utiliserons trois boutons de type "radio". Chacun d'entre eux représentera un des trois formats de date supportés par le web service. Quand l'utilisateur clique sur un des boutons "radio", le navigateur exécutera le code JavaScript ci-dessous :

```
$find("#mcd1_dpe1").populate(this.value)
```

Ce code accède au **DynamicPopulateExtender** (ne vous inquiétez pas pour cet indientifiant étrange, ce sera résolu plus tard) et déclenchera le remplissage automatique avec les données. Dans le contexte des boutons "radio", `this.value` fait référence à leur valeur format1, format2 or format3, qui est exactement la valeur requise par la méthode web.

L'unique chose manquante est le **DynamicPopulateExtender** qui liera les boutons "radio" au web service.

```
<ajaxToolkit:DynamicPopulateExtender ID="dpe1" runat="server"  
  ClearContentsDuringUpdate="true"  
  TargetControlID="mcd1$myDate"  
  ServicePath="DynamicPopulate.cs.aspx"  
  ServiceMethod="getDate"/>
```

Encore une fois, vous avez peut être noté l'identifiant étrange dans le contrôle: **mcd1\$myDate** à la place de **myDate**. Auparavant, le code JavaScript utilisait **mcd1\_dpe1** pour accéder au **DynamicPopulateExtender** à la place de **dpe1**. Cette stratégie de nommage est requise lorsqu'on utilise un **DynamicPopulateExtender** au sein d'un contrôle. De plus, vous devez embarquer votre contrôle utilisateur d'une certaine façon pour qu'il fonctionne. Créons une nouvelle page ASP.NET et référençons notre contrôle utilisateur implémenté avec le tag suivant :

```
<%@ Register TagPrefix="uc1" TagName="myCustomDate" Src="~/DynamicPopulate2.cs.ascx"%>
```

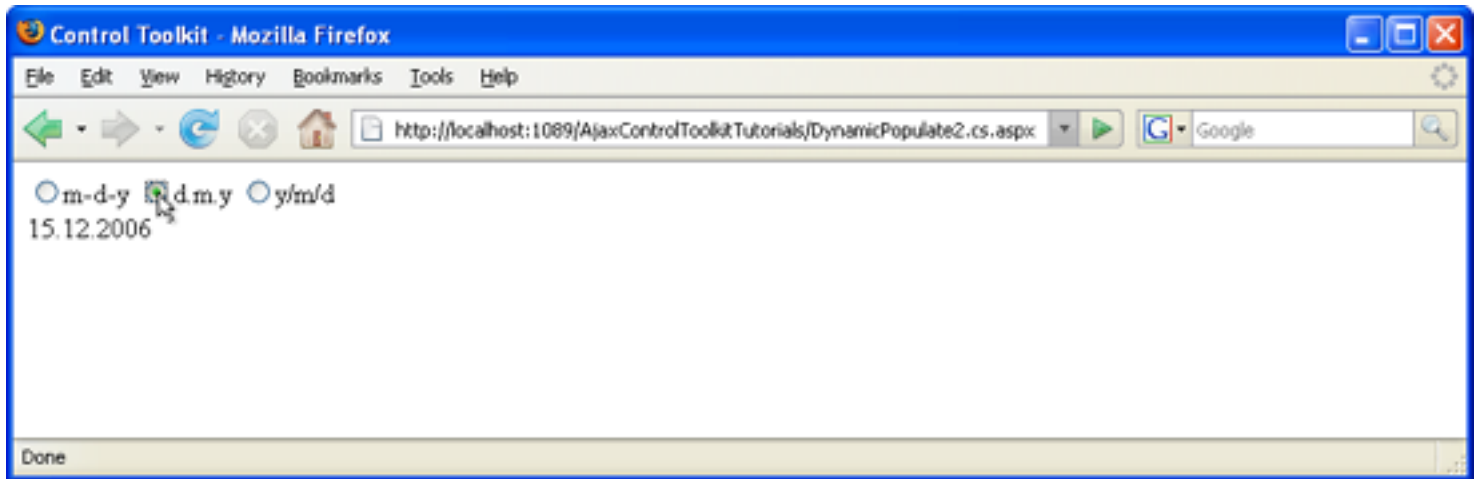
Ensuite, incluons l'ASP.NET Ajax **ScriptManager** dans notre page:

```
<asp:ScriptManager ID="asm" runat="server" />
```

Enfin, ajoutons le contrôle utilisateur à la page. Vous n'avez de définir que l'attribut ID (et `runat="server"` bien sûr), mais vous devez aussi lui donner un nom spécifique: **mcd1**  **since** qui est le préfixe utilisé dans le contrôle utilisateur pour que le code JavaScript y accède.

```
<div>  
  <uc1:myCustomDate ID="mcd1" runat="server" />  
</div>
```

Et c'est tout! La page se comporte comme prévu : l'utilisateur clique sur un bouton "radio", le contrôle dans le ToolKit appelle le web service et affiche la date courante au format désiré.



*Les boutons résident au sein du contrôle utilisateur*