



Sitecore CMS 6

Data Definition API

Cookbook

A Conceptual Overview for CMS Developers

Table of Contents

Chapter 1	Introduction.....	3
Chapter 2	Data Templates	4
2.1	How to Access a Data Template.....	5
2.1.1	How to Access the Data Template Associated with an Item	5
2.1.2	How to Access an Existing Data Template	5
2.1.3	How to Create a Data Template.....	5
2.1.4	How to Access the Icon of a Data Template	6
2.1.5	How to Access the Base Templates of a Data Template	6
2.2	How to Access a Data Template Section.....	7
2.2.1	How to Access an Existing Data Template Section	7
2.2.2	How to Add a Section to a Data Template	7
2.2.3	How to Access the Icon of a Data Template Section.....	8
2.2.4	How to Access the Sort Order of a Data Template Section.....	8
2.3	How to Access a Data Template Field.....	9
2.3.1	How to Access an Existing Data Template Field	9
2.3.2	How to Add a Field to a Data Template Section	9
2.3.3	How to Access the Properties of a Data Template Field	10
	How to Access the Sort Order of a Data Template Field	10
2.4	How to Access the Standard Values of a Data Template.....	11
2.4.1	How to Set a Standard Value of a Data Template Field	11
	How to Specify a Default Workflow for Items Based on a Data Template.....	11
	How to Specify Default Layout Details for Items Based on a Data Template.....	12
	How to Specify Default Insert Options for Items Based on a Data Template	12
	How to Specify a Default Subitems Sorting Rule for Items Based on a Data Template.....	12
Chapter 3	Branch Templates	14
3.1	How to Access an Existing Branch Template	15
3.2	How to Create a Branch Template.....	16
3.2.1	How to Create a Branch Template Containing a Single Item	16
3.3	How to Add an Item to a Branch Template	17
3.3.1	How to Copy Items to a Branch Template	17
3.4	How to Access Field Values in Items in Branch Templates.....	18

Chapter 1

Introduction

This Data Definition API Cookbook provides a conceptual overview of Application Programming Interfaces (APIs) that Sitecore developers can use to access and manipulate data template and branch template definitions.¹

This document contains the following chapters:

- Chapter 1 – Introduction
- Chapter 2 – Data Templates
- Chapter 3 – Branch Templates

¹ For more information about the topics described in this document, see the Data Definition Reference manual at <http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Reference.aspx> and the Data Definition Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Cookbook.aspx>. For more information about specific Sitecore APIs, see the Sitecore API reference at http://sdn.sitecore.net/Reference/Sitecore%206/Sitecore_6_API_Reference.aspx.

Chapter 2

Data Templates

This chapter provides information about and examples of APIs that you can use to access and manipulate data template, section, and field definitions.

This chapter contains the following sections:

- How to Access a Data Template
- How to Access a Data Template Section
- How to Access a Data Template Field
- How to Access the Standard Values of a Data Template

2.1 How to Access a Data Template

You can use the `Sitecore.Data.Items.TemplateItem` class to access data template definitions.

2.1.1 How to Access the Data Template Associated with an Item

You can use the `Sitecore.Data.Items.Item.Template` property to access the data template associated with an item.² For example, to access the data template associated with the `/Sitecore/Content/Home` item in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.TemplateItem template = home.Template;
```

2.1.2 How to Access an Existing Data Template

You can use the `Sitecore.Data.Database.Templates` collection property to access an existing data template. For the collection key, use the GUID of the data template, or the path to the data template definition item relative to the `/Sitecore/Templates` item, without the leading slash character (“/”). For example, to access the `Sample/Sample` Item data template in the Master database (`/Sitecore/Templates/Sample/Sample` Item):

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];

if (sample==null)
{
    //TODO: handle case that data template does not exist
}
```

The `Sitecore.Data.Database.Templates` collection returns `Null` if the specified data template does not exist.

2.1.3 How to Create a Data Template

You can use the `Sitecore.Data.Database.Templates.CreateTemplate()` method to create a data template. You must specify the data template name and the existing parent folder to contain the data template. For example, to ensure the `User Defined/Example Data Template` data template (`/Sitecore/Templates/User Defined/Example Data Template`) exists in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];

if (example == null)
{
    Sitecore.Data.Items.Item parent = master.GetItem("/sitecore/templates/user defined");
    example = master.Templates.CreateTemplate("Example Data Template", parent);
}
```

² For more information about the `Sitecore.Data.Items.Item`, `Sitecore.Data.Database`, and other APIs used in this section and throughout this document, see the [Content API Cookbook](http://sdn.sitecore.net/Reference/Sitecore%206/Content%20API%20Cookbook.aspx) at <http://sdn.sitecore.net/Reference/Sitecore%206/Content%20API%20Cookbook.aspx>.

2.1.4 How to Access the Icon of a Data Template

You can use the `Sitecore.Data.Items.TemplateItem.InnerItem.Appearance.Icon` property to access the icon of a data template.³ For example, to set the icon of the `Sample/Sample Item` data template in the Master database to `network/16x16/home.png`:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
sample.InnerItem.Editing.BeginEdit();
sample.InnerItem.Appearance.Icon = "network/16x16/home.png";
sample.InnerItem.Editing.EndEdit();
```

2.1.5 How to Access the Base Templates of a Data Template

You can use the `Sitecore.Data.Items.TemplateItem.BaseTemplates` property to access the base templates of a data template. You can update the base templates associated with a data template by setting the value of the `Sitecore.FieldIDs.BaseTemplate` field in the data template definition item. For example, to ensure that the `User Defined/Example Data Template` data template in the Master database includes the `Sample/Sample Item` and `Common/Folder` base templates:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Fields.MultilistField baseTemplates =
    example.InnerItem.Fields[Sitecore.FieldIDs.BaseTemplate];

foreach (string key in new string[] { "sample/sample item",
    Sitecore.TemplateIDs.Folder.ToString() })
{
    Sitecore.Data.Items.TemplateItem template = master.Templates[key];

    if (!baseTemplates.Contains(template.ID.ToString()))
    {
        example.InnerItem.Editing.BeginEdit();
        baseTemplates.Add(template.ID.ToString());
        example.InnerItem.Editing.EndEdit();
    }
}
```

³ For more information about data template icons, see the Client Configuration Reference manual at <http://sdn.sitecore.net/Reference/Sitecore%206/Client%20Configuration%20Reference.aspx> and the Client Configuration Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206/Client%20Configuration%20Cookbook.aspx>.

2.2 How to Access a Data Template Section

You can use the `Sitecore.Data.Items.TemplateSectionItem` class to access data template section definitions.

2.2.1 How to Access an Existing Data Template Section

You can use the `Sitecore.Data.Items.TemplateItem.GetSection()` method to access an existing data template section. For example, to access the `Data` section in the `Sample/Sample Item` data template in the `Master` database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.TemplateSectionItem data = sample.GetSection("data");
```

If the data template definition item contains the specified section, the `Sitecore.Data.Items.TemplateItem.GetSection()` method returns that section. Otherwise, the `Sitecore.Data.Items.TemplateItem.GetSection()` method returns the first matching section encountered while recursively processing the base templates of that data template. If none of the base templates contain the section, then the `Sitecore.Data.Items.TemplateItem.GetSection()` method returns `Null`.

You can differentiate between sections defined within a data template and sections defined within one of its base templates by comparing the ID of the parent of a data template section definition item to the ID of a data template definition item as demonstrated in the following section [How to Add a Section to a Data Template](#).

2.2.2 How to Add a Section to a Data Template

You can use the `Sitecore.Data.Items.TemplateItem.AddSection()` method to add a section to a data template. For example, to ensure the `Data` section exists in the `User Defined/Example Data Template` data template in the `Master` database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.TemplateSectionItem data = example.GetSection("data");

if (data==null || data.InnerItem.Parent.ID!=example.ID)
{
    data = example.AddSection("Data", false);
}
```

Note

If you do not specify `False` as the second parameter to the `Sitecore.Data.Items.TemplateItem.AddSection()` method, the method may return a data template section defined in a base template.

2.2.3 How to Access the Icon of a Data Template Section

You can use the `Sitecore.Data.Items.TemplateItem.InnerItem.Appearance.Icon` property to access the icon associated with a data template section.⁴ For example, to set the icon of the Data section in the `Sample/Sample Item` data template in the Master database to `applications/16x16/folder.png`:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.TemplateSectionItem data = sample.GetSection("data");
data.InnerItem.Editing.BeginEdit();
data.InnerItem.Appearance.Icon = "applications/16x16/folder.png";
data.InnerItem.Editing.EndEdit();
```

2.2.4 How to Access the Sort Order of a Data Template Section

You can use the `Sitecore.Data.Items.TemplateSectionItem.Sortorder` property to access the sort order of a data template section. The Content Editor sorts the sections within each data template according to the values of the sort order property. If you do not sort the sections, Sitecore sorts them alphabetically by name.

You can update the `Sitecore.FieldIDs.Sortorder` field in the `Sitecore.Data.Items.TemplateSectionItem.InnerItem.Fields` collection property to set the sort order for a data template section. For example, to set the sort order property of the Data section in the `Sample/Sample Item` data template in the Master database to 10:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.TemplateSectionItem data = sample.GetSection("data");

if (data.Sortorder!=10)
{
    data.InnerItem.Editing.BeginEdit();
    data.InnerItem.Fields[Sitecore.FieldIDs.Sortorder].Value = 10.ToString();
    data.InnerItem.Editing.EndEdit();
}
```

⁴ For more information about data template section icons, see the Client Configuration Reference manual and the Client Configuration Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206.aspx>.

2.3 How to Access a Data Template Field

You can use the `Sitecore.Data.Items.TemplateFieldItem` class to access data template field definitions.

2.3.1 How to Access an Existing Data Template Field

You can use the `Sitecore.Data.Items.TemplateItem.GetField()` method to access a data template field when the section containing the field is unimportant. For example, to access the Title field in the User Defined/Example Data Template data template in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.TemplateFieldItem title = sample.GetField("title");
```

Note

The `Sitecore.Data.Items.TemplateItem.GetField()` method may return a field from a base template.

2.3.2 How to Add a Field to a Data Template Section

You can use the `Sitecore.Data.Items.TemplateSectionItem.AddField()` method to add a field to a data template section. For example, to ensure the Title field exists in the Data section of the User Defined/Example Data Template data template in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.TemplateSectionItem data = example.GetSection("data");

if (data == null || data.InnerItem.Parent.ID != example.ID)
{
    data = example.AddSection("Data", false);
}

Sitecore.Data.Items.TemplateFieldItem title = data.GetField("title");

if (title == null)
{
    Sitecore.Data.Items.TemplateFieldItem field = data.AddField("Title");
}
```

Note

If the data template section definition item does not contain the specified field, the `Sitecore.Data.Items.TemplateSectionItem.GetField()` method returns Null. The `Sitecore.Data.Items.TemplateSectionItem.GetField()` method does not process base templates.

2.3.3 How to Access the Properties of a Data Template Field

You can use the following properties of the `Sitecore.Data.Items.TemplateFieldItem` class to access the properties of a data template field:⁵

- `Sitecore.Data.Items.TemplateFieldItem.Source`: The configuration source of the field.
- `Sitecore.Data.Items.TemplateFieldItem.Title`: The title of the field.
- `Sitecore.Data.Items.TemplateFieldItem.Type`: The data type of the field.

You can access additional properties of a data template using the `Sitecore.Data.Items.TemplateFieldItem.InnerItem.Fields` collection property. For the collection key, specify a member of the `Sitecore.TemplateFieldIDs` class.

For example, to set the data type, source, title, and description properties for the Title field in the `Sample/Sample Item` data template in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.TemplateFieldItem title = sample.GetField("title");
Sitecore.Data.Database core = Sitecore.Configuration.Factory.GetDatabase("core");
Sitecore.Data.Items.Item textFieldType =
    core.GetItem("/sitecore/system/field types/simple types/single-line text");
title.BeginEdit();
title.Type = textFieldType.Name;
title.Source = "//TODO: replace with field source property";
title.Title = "//TODO: replace with field title";
title.InnerItem.Fields[Sitecore.TemplateFieldIDs.Description].Value =
    "//TODO: replace with field description";
title.EndEdit();
```

How to Access the Sort Order of a Data Template Field

You can use the `Sitecore.Data.Items.TemplateFieldItem.Sortorder` property to access the sort order of a data template field. The Content Editor sorts the fields within each section according to the values of the sort order property of the fields. If you do not sort the fields, Sitecore sorts them automatically alphabetically by name. For example, to sort the Title field in the `Sample/Sample Item` data template in the Master database after the Title field, assuming they exist in the same section:

```
Sitecore.Data.Database master =
    Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem sample = master.Templates["sample/sample item"];
Sitecore.Data.Items.TemplateFieldItem title = sample.GetField("title");
Sitecore.Data.Items.TemplateFieldItem text = sample.GetField("text");
title.BeginEdit();
title.Sortorder = 100;
title.EndEdit();
text.BeginEdit();
text.Sortorder = 200;
text.EndEdit();
```

⁵ For more information about data template field properties, see the Data Definition Reference manual, the Client Configuration Reference manual, and the Client Configuration Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206.aspx>.

2.4 How to Access the Standard Values of a Data Template

You can use the `Sitecore.Data.Items.TemplateItem.StandardValues` property to access the standard values of a data template. If the `Sitecore.Data.Items.TemplateItem.StandardValues` property is Null, then standard values do not exist for the data template.

If standard values do not exist for a data template, you can use the `Sitecore.Data.Items.TemplateItem.CreateStandardValues()` method to create the standard values item for that data template. For example, to access the standard values for the `User Defined/Example Data Template` data template in the Master database, creating the standard values item if needed:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.Item standardValues = example.StandardValues;

if (standardValues == null)
{
    standardValues = example.CreateStandardValues();
}
```

2.4.1 How to Set a Standard Value of a Data Template Field

You can use the `Sitecore.Data.Items.Item.Fields` collection property to set the standard value for a data template field.⁶ You can include tokens such as `$name` and `$now` in the standard value of a field.⁷ For example, to set the standard value for the `Title` field in the `User Defined/Example Data Template` data template in the Master database to `$name`:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.Item standardValues = example.StandardValues;
standardValues.Editing.BeginEdit();
standardValues.Fields["title"].Value = "$name";
standardValues.Editing.EndEdit();
```

How to Specify a Default Workflow for Items Based on a Data Template

You can set the value of the `Sitecore.FieldIDs.DefaultWorkflow` field in the standard values for a data template to specify a default workflow for all items based on that data template. For example, to set the default workflow for the `User Defined/Example Data Template` data template in the Master database to the `/Sitecore/System/Workflows/Sample Workflow` workflow:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.Item standardValues = example.StandardValues;
Sitecore.Data.Items.Item sampleWorkflow =
    master.GetItem("/sitecore/system/workflows/sample workflow");
standardValues.Editing.BeginEdit();
standardValues.Fields[Sitecore.FieldIDs.DefaultWorkflow].Value =
    sampleWorkflow.ID.ToString();
standardValues.Editing.EndEdit();
```

⁶ For more information about setting field values, see the Content API Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206/Content%20API%20Cookbook.aspx>.

⁷ For more information about the tokens you can use in standard values, see the Data Definition Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206/Data%20Definition%20Cookbook.aspx>.

How to Specify Default Layout Details for Items Based on a Data Template

You can set the value of the `Sitecore.FieldIDs.LayoutField` field in the standard values of a data template to specify default layout details for all items based on that data template. For example, to copy layout details from the `/Sitecore/Content/Home` item to the standard values of the `User Defined/Example Data Template` data template in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.Item standardValues = example.StandardValues;
standardValues.Editing.BeginEdit();
standardValues.Fields[Sitecore.FieldIDs.LayoutField].Value =
    home.Fields[Sitecore.FieldIDs.LayoutField].Value;
standardValues.Editing.EndEdit();
```

How to Specify Default Insert Options for Items Based on a Data Template

You can set the values of the `Sitecore.FieldIDs.Branches` and `__Insert Rules` fields in the standard values of a data template to specify default insert options for all items based on that data template. For example, to ensure insert options for the `User Defined/Example Data Template` data template in the Master database include the `Sample/Sample Item` and `Common/Folder` data templates, and that insert rules include those defined in the `/Sitecore/Content/Home` item:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.Item standardValues = example.StandardValues;
standardValues.Editing.BeginEdit();
Sitecore.Data.Fields.MultilistField standardInsertOptions =
    standardValues.Fields[Sitecore.FieldIDs.Branches];

foreach (string templateKey in new string[] { "sample/sample item",
    Sitecore.TemplateIDs.Folder.ToString() })
{
    Sitecore.Data.Items.TemplateItem template = master.Templates[templateKey];

    if (!standardInsertOptions.Contains(template.ID.ToString()))
    {
        standardInsertOptions.Add(template.ID.ToString());
    }
}

Sitecore.Data.Fields.MultilistField standardInsertRules =
    standardValues.Fields[" Insert Rules"];
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
Sitecore.Data.Fields.MultilistField homeInsertRules = home.Fields[" Insert Rules"];

foreach(string homeInsertRuleId in homeInsertRules.Items)
{
    if (!standardInsertRules.Contains(homeInsertRuleId))
    {
        standardInsertRules.Add(homeInsertRuleId);
    }
}

standardValues.Editing.EndEdit();
```

How to Specify a Default Subitems Sorting Rule for Items Based on a Data Template

You can set the value of the `Sitecore.FieldIDs.SubitemsSorting` field in the standard values for a data template to specify the default subitem sorting rule for all items based on the data template. For

example, to set the default subitem sorting rule for the User Defined/Example Data Template data template in the Master database to the /Sitecore/System/Settings/Subitems Sorting/Updated subitems sorting rule:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Database core = Sitecore.Configuration.Factory.GetDatabase("core");
Sitecore.Data.Items.Item rule =
    core.GetItem("/sitecore/system/settings/subitems sorting/updated");
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.TemplateItem example =
    master.Templates["user defined/example data template"];
Sitecore.Data.Items.Item standardValues = example.StandardValues;
standardValues.Editing.BeginEdit();
standardValues.Fields[Sitecore.FieldIDs.SubitemsSorting].Value = rule.ID.ToString();
standardValues.Editing.EndEdit();
standardValues.Editing.EndEdit();
```

Chapter 3

Branch Templates

This chapter contains information about and examples of APIs that you can use to access and manipulate data templates and branch templates.

This chapter contains the following sections:

- How to Access an Existing Branch Template
- How to Create a Branch Template
- How to Add an Item to a Branch Template
- How to Access Field Values in Items in Branch Templates
- Error! Reference source not found.

3.1 How to Access an Existing Branch Template

You can use the `Sitecore.Data.Items.BranchItem` class to access branch templates. You can use the `Sitecore.Data.Database.Branches` collection property to access an existing branch template. For the key to the collection, use the path to the branch template definition item relative to the `/Sitecore/Templates` item, without the leading slash character ("/"). For example, to access the `Branches/User Defined/Example Branch Template` branch template (`/Sitecore/Templates/Sitecore/Branches/User Defined/Example Branch Template`) in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.BranchItem exampleBranch =
    master.Branches["branches/user defined/example branch template"];

if (exampleBranch==null)
{
    //TODO: handle case that branch template does not exist
}
```

3.2 How to Create a Branch Template

You can use the `Sitecore.Data.Items.Item.Add()` method to create a branch template. For example, to ensure the `Branches/User Defined/Example Branch Template` branch template exists in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.BranchItem exampleBranch =
    master.Branches["branches/user defined/example branch template"];

if (exampleBranch == null)
{
    Sitecore.Data.Items.Item parent =
        master.GetItem("/sitecore/templates/branches/user defined");
    Sitecore.Data.Items.TemplateItem example =
        master.Templates[Sitecore.TemplateIDs.BranchTemplate];
    exampleBranch = new Sitecore.Data.Items.BranchItem(
        parent.Add("Example Branch Template", example));
}
```

3.2.1 How to Create a Branch Template Containing a Single Item

You can use the `Sitecore.Data.Masters.CreateMaster()` method to create a branch template containing a single item named `$name`. You may want to copy the icon from the data template associated with the item to the branch template definition item. For example, to create the `Branches/User Defined/Sample Item` branch template containing a single item named `$name` based on the `Sample/Sample Item` data template in the Master database, copying the icon from the data template to the branch template definition item:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item parent =
    master.GetItem("/sitecore/templates/branches/user defined");
Sitecore.Data.Items.TemplateItem exampleTemplate =
    master.Templates["sample/sample item"];
Sitecore.Data.Items.Item exampleItem =
    Sitecore.Data.Masters.Masters.CreateMaster(parent, exampleTemplate.ID);
Sitecore.Data.Items.BranchItem exampleBranch =
    new Sitecore.Data.Items.BranchItem(exampleItem);
exampleBranch.InnerItem.Editing.BeginEdit();
exampleBranch.InnerItem.Appearance.Icon = exampleTemplate.Icon;
exampleBranch.InnerItem.Editing.EndEdit();
```


3.3 How to Add an Item to a Branch Template

You can use the `Sitecore.Data.Items.Item.Add()` method to add an item to a branch template. For example, to add an item named `$name` based on the `Sample/Sample Item` data template to the `Branches/User Defined/Example Branch Template` branch template in the Master database if that item does not already exist:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.Item exampleBranch =
    master.Branches["branches/user defined/example branch template"];
Sitecore.Data.Items.Item child = exampleBranch.Children["$name"];

if (child == null)
{
    Sitecore.Data.Items.TemplateItem sampleTemplate =
        master.Templates["sample/sample item"];
    child = exampleBranch.Add("$name", sampleTemplate);
}
```

3.3.1 How to Copy Items to a Branch Template

You can use the `Sitecore.Data.Items.Item.CopyTo()` method to copy an existing item or hierarchy of items to a branch template. For example, to copy the `/Sitecore/Content/Home` item to the `Branches/User Defined/Sample Item` branch template in the Master database:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.BranchItem exampleBranch =
    master.Branches["branches/user defined/sample item"];
Sitecore.Data.Items.Item sampleItem = exampleBranch.InnerItem.Children["$name"];
Sitecore.Data.Items.Item home = master.GetItem("/sitecore/content/home");
home.CopyTo(exampleBranch.InnerItem, home.Name);
```

3.4 How to Access Field Values in Items in Branch Templates

You can use the `Sitecore.Data.Items.Item.Fields` collection property access field values in the items of a branch template.⁸ For example, to set the value of the Title field in the child named `$name` beneath under the `Branches/User Defined/Example Branch Template` branch template in the Master database to `$name`:

```
Sitecore.Data.Database master = Sitecore.Configuration.Factory.GetDatabase("master");
Sitecore.Data.Items.BranchItem sampleBranch =
    master.Branches["branches/user defined/example branch template"];
Sitecore.Data.Items.Item name = sampleBranch.InnerItem.Children["$name"];
name.Editing.BeginEdit();
name.Fields["title"].Value = " $name";
name.Editing.EndEdit();
```

⁸ For more information about setting field values, see the Content API Cookbook at <http://sdn.sitecore.net/Reference/Sitecore%206/Content%20API%20Cookbook.aspx>.