

Article

[Kurro Lopez](#) · Aug 4, 2021 4m read

## Add a business item in your production by code

Hi community,

This is another article about how to perform actions that you can do in the web portal but via code.

Today.... Add a business item in your production by code

### Introduction

If you want to add a new business item (business service, business process or business operation), you usually do it from the production configuration.



# Welcome, Kurro Lopez

View:

<b>Home</b>	<b>Configure</b> >	<b>Production</b> ⓘ
	<b>Build</b> >	<b>Business Partners</b> ⓘ
	<b>View</b> >	<b>Credentials</b> ⓘ
	<b>List</b> >	<b>Schedule Specs</b> ⓘ
	<b>Monitor</b> >	<b>Data Lookup Tables</b> ⓘ
	<b>Manage</b> >	<b>System Default Settings</b> ⓘ
<b>Interoperability</b>	<b>Interoperate</b> >	<b>Purge Data Settings</b> ⓘ
	<b>Test</b> >	<b>Enterprise Systems</b> ⓘ
<b>System Operation</b>		Public-Service Registry
		<b>External-Service Registry</b> ⓘ
<b>System Explorer</b>		<b>Message Bank Link</b> ⓘ
<b>System Administration</b>		

Interoperability > Production Configuration - (Test.MyProduction)

## Production Configuration

Start Stop Refresh Sort: Name Status Number View:

Production Stopped Category: All Legend Production Settings

Services Processes Operations

### BUSINESS SERVICE WIZARD

Add a new Business Service to this Production.

All Services X12 Input Business Metric

Service Class:

Service Name:

Display Category:

Comment:

Enable Now

Use this form to add a new business service to the production.  
For help with any setting in this form, hover the cursor over the setting name.

Cancel OK

By the same way, if you are not able to access to this portal, you can add your business item into your production

by code.

## AddBusinessItem

```

/// Add a item into the production
/// <ul>
/// <li><var>pProduction</var> Name of the production.</li>
/// <li><var>pName</var> Set the name of the item. This value is mandatory.</li>
/// <li><var>pClassName</var> Set the classname of the item. This value is mandatory.
</li>
/// <li><var>pCategory</var> Set the name of the category into the production.</li>
/// <li><var>pEnable</var> Set if the item is enabled.</li>
/// <li><var>pLogTraceEvents</var> Set event traces.</li>
/// <li><var>pPoolSize</var> Pool size.</li>
/// <li><var>pComment</var> Comment.</li>
/// </ul>
/// <example>Do myClass.AddItem("myPath.MyProduction", Item1, "myPath.MyClass", "Int
ernal")</example>
ClassMethod AddBusinessItem(pProduction As %String, pName As %String, pClassName As %
String, pCategory As %String = "", pEnabled As %Boolean = 1, pLogTraceEvents As %Bool
ean = 1, pPoolSize As %Integer = 1, pComment As %String = "") As %Status
{
    Set ret = $$$OK
    Try {
        // Check the mandatories values
        If ((pName = "") && (pClassName = ""))
        {
            If '##class(Ens.Config.Item).NameExists(pProduction,pName,.idItem)
            {
                Set produccion = ##class(Ens.Config.Production).%OpenId(pProduction)
                Write !,"Installing "_pName
                Set objItem = ##class(Ens.Config.Item).%New()
                Set objItem.Name = pName
                Set objItem.ClassName = pClassName
                Set objItem.Enabled = pEnabled
                Set objItem.LogTraceEvents = pLogTraceEvents
                Set objItem.Category = pCategory
                Set objItem.PoolSize = pPoolSize
                Set objItem.Comment = pComment

                Do produccion.Items.Insert(objItem)
                Set st = produccion.%Save()
                Kill objItem
                Do ##class(Ens.Director).RestartProduction(0,1)
                Write !,"Item "_pName_" installed."
            }
            Else
            {
                Write !,"Item "_pName_" already exists."
            }
        }
    }
    Else
    {
        If (pName = "") set attrib = "pName"
        If (pClassName = "") set attrib = "pClassName"

        $$$ThrowOnError($System.Status.Error(406,attrib))
    }
}

```

```

    }
  }
  Catch ex {
    Set ret = ex.AsStatus()
    Write !,"Error #",$System.Status.GetErrorCodes(ret),!
    Write !,$System.Status.GetOneStatusText(ret,1),!
  }

  Quit ret
}

```

if you want to add a business service, your class should be inherited from `Ens.BusinessService` (as always)

## RemoveBusinessItem

```

/// Remove a business item from a production
/// <ul>
/// <li><var>pProduction</var> Name of the production.</li>
/// <li><var>pName</var> Name of the business item to remove. This value is mandatory
/// </li>
/// </ul>
/// <example>Do myClass.RemoveItem("Item1")</example>
ClassMethod RemoveBusinessItem(pProduction As %String, pName As %String) As %Status
{
  Set ret = $$$OK

  Try {

    // Validate mandatory values
    If (pName '=')
    {
      If ##class(Ens.Config.Item).NameExists(pProduction,pName,.idItem)
      {
        Set produccion = ##class(Ens.Config.Production).%OpenId(pProduction)
        Write !,"Removing "_pName
        Set objItem = produccion.FindItemByConfigName(pName)
        Do produccion.RemoveItem(objItem)
        Set st = produccion.%Save()
        Kill objItem
        Do ##class(Ens.Director).RestartProduction(0,1)
        Write !,"Item "_pName_" removed"
      }
      Else
      {
        Write !,"Item "_pName_" not found."
      }
    }
    Else
    {
      If (pName = '') set attrib = "pName"

      $$$ThrowOnError($System.Status.Error(406,attrib))
    }
  }
  Catch ex {
    Set ret = ex.AsStatus()
    Write !,"Error #",$System.Status.GetErrorCodes(ret),!
    Write !,$System.Status.GetOneStatusText(ret,1),!
  }
}

```

```
    }  
    Quit ret  
}
```

After a remove item, it restarts automatically the production to refresh the content.

If you add an item, you can configure its parameters using the default configuration, but it will be other post ;)

Best regards,  
Kurro Lopez

[#Beginner](#) [#Business Operation](#) [#Business Service](#) [#ObjectScript](#) [#Tips & Tricks](#) [#Caché](#) [#Ensemble](#)  
[#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/add-business-item-your-production-code>