

---

Question

[Alexey Maslov](#) · Apr 1, 2017

## %Installer Manifest: How to partially reconfigure the NameSpace (add some subscript level mapping)?

I've got an issue trying to write a code which should add some global mapping to already existing namespace. Here is its skeleton:

```
Include %occInclude

/// Create simple global mapping (main DB and cachetemp only)
Class %z.Mapping Extends %Library.RegisteredObject
{

    /// Application Definition
    XData qMS [ XMLNamespace = INSTALLER ]
    {
        <Manifest>
        <If Condition='$L("${NAMESPACE}")=0'>
        <!-- Report an error if the namespace wasn't specified -->
        <Error Status="$$$NamespaceDoesNotExist">
        <Arg Value="${NAMESPACE}" />
        </Error>
        </If>

        <If Condition='$zcv("${NAMESPACE}","U")=%SYS'>
        <!-- Report an error if the namespace = %SYS -->
        <Error Status="$$$GeneralError">
        <Arg Value="${NAMESPACE} should not be %SYS" />
        </Error>
        </If>

        <Invoke Class="%z.Mapping" Method="GetMainDB" CheckStatus="0" Return="MainDB" >
        <Arg Value="${NAMESPACE}" />
        </Invoke>

        <If Condition='$L("${MainDB}")=0'>
        <!-- Report an error if the globals database was not detected -->
        <Error Status="$$$GeneralError">
        <Arg Value="${NAMESPACE} has no Globals database" />
        </Error>
        </If>

        <Var Name="DBRESOURCE" Value="%DB_%DEFAULT" />

        <Namespace Name="${NAMESPACE}" Create="no" >

        <Log Level="0" Text=
        "${NAMESPACE} is ready to be configured using ${MainDB} and ${DBRESOURCE}" />
    }
}
```

```
<Configuration>

<!-- ??? Configure the database that should exist upto this step -->
  <Database Name="{MainDB}"
    Dir="{MainDB}"
    Create="no"
    Resource="{DBRESOURCE}"
    PublicPermissions=""
  />
<!-- ??? Configure the database that should exist upto this step -->

  <GlobalMapping Global="Q(&quot;0!&quot;):(END)" From="CACHETEMP"/>
  <GlobalMapping Global=
"Q(&quot;0mseSTATUS&quot;):( &quot;0mseSTATUS~&quot;)" From="{MainDB}"/>
  <GlobalMapping Global="Qa" From="CACHETEMP"/>
  <GlobalMapping Global="Qi" From="CACHETEMP"/>
  <GlobalMapping Global="Qi(&quot;MO&quot;)" From="{MainDB}"/>

</Configuration>

</Namespace>

</Manifest>
}

/// This is a method generator whose code is generated by XGL.
ClassMethod setup(ByRef pVars, pLogLevel As %Integer, pInstaller As
%Installer.Installer, pLogger As %Installer.AbstractLogger) As %Status [
CodeMode = objectgenerator, Internal ]
{
#; Let our XGL document generate code for this method.
Quit ##class(%Installer.Manifest).%Generate(%compiledclass, %code, "qMS")
}

ClassMethod GetMainDB(pNamespace) As %String
{
new $namespace s $namespace="%SYS"
s sc=##class(Config.Namespaces).Get(pNamespace,.p)
if 'sc d $system.OBJ.DisplayError(sc) quit ""
quit p("Globals")
}

/// Invoke the installer passing in some variables
///
zn "USER1" s sc=$system.OBJ.Load("mapping.xml","ck") d:'sc $system.Status.DisplayError(sc)
///
s sc=##class(%z.Mapping).RunInstallWithLog($zu(12)_"mapping.log",2) d:'sc $system.Status.DisplayError(sc)
///
ClassMethod RunInstallWithLog(pLogfile As %String, pLogLevel = 1) As %Status
{
#dim tVars
#dim tStatus As %Status

Set tVars("NAMESPACE") = $zu(5)

// Construct a file logger
```

```
#dim tLogger As %Installer.FileLogger = ##class(
%Installer.FileLogger).%New(1,pLogfile)

// Invoke the installer
Set tStatus = ..setup(.tVars,pLogLevel,,tLogger)
Do:$$$ISERR(tStatus) $system.OBJ.DisplayError(tStatus)
Quit tStatus
}

}
```

I've run it with following commands:

```
zn "USER1" s sc=##class(%z.Mapping).RunInstallWithLog($zu(12)_"mapping.log",2) d:'sc
$system.Status.DisplayError(sc)
```

And it worked. But if I'm excluding an [unneeded] code fragment that resides between two lines:

<!-- ??? Configure the database that should exist upto this step -->

I'm getting several errors, please see below. Why this fragment is still needed nevertheless I marked my namespace with **Create="no"** attribute? The same bugs are thrown with an option **Create="overwrite"**.

```
?2017-03-31 19:24:51 0 %z.Mapping: Installation starting at 2017-03-31 19:24:51, LogL
evel=2
2017-03-31 19:24:51 0 : USER1 is ready to be configured using USER1 and %DB_%DEFAULT
2017-03-31 19:24:51 1 CreateNamespace: Creating namespace USER1 using /
2017-03-31 19:24:51 2 CreateNamespace: Modifying namespace USER1
2017-03-31 19:24:51 0 %z.Mapping: ERROR #5002: Cache error: <SUBSCRIPT>%LoadData+6^Co
nfig.Databases.1 ^SYS("CONFIG","CACHE","Databases","")
ERROR #5659: Property 'Config.Namespaces::Globals(3@Config.Namespaces,ID=CACHE| |Names
paces| |USER1)' required
ERROR #7202: Datatype value '' length less than MINLEN allowed of 1
ERROR #5659: Property 'Config.Namespaces::Routines(3@Config.Namespaces,ID=CACHE| |Name
spaces| |USER1)' required
ERROR #7202: Datatype value '' length less than MINLEN allowed of 1
2017-03-31 19:24:51 0 %z.Mapping: ERROR #ConfigFailed: Unknown status code: <Ins>Conf
igFailed(,,,,,,)
> ERROR #5002: Cache error: <SUBSCRIPT>%LoadData+6^Config.Databases.1 ^SYS("CONFIG"
,"CACHE","Databases","")
> ERROR #5659: Property 'Config.Namespaces::Globals(3@Config.Namespaces,ID=CACHE| |N
amespaces| |USER1)' required
> ERROR #7202: Datatype value '' length less than MINLEN allowed of 1
> ERROR #5659: Property 'Config.Namespaces::Routines(3@Config.Namespaces,ID=CACHE| |
Namespaces| |USER1)' required
> ERROR #7202: Datatype value '' length less than MINLEN allowed of 1
2017-03-31 19:24:51 0 %z.Mapping: Installation failed at 2017-03-31 19:24:51
2017-03-31 19:24:51 0 %Installer: Elapsed time .178622s
```

The error report shows that %Installer still tries to configure database using empty internal structures. So it seems that I have two options:

1 - Write getters similar to GetMainDB for other database attributes (Dir, Resource, etc). Why not one getter for all of them? Just because there is no option to pass arguments by reference, so only one value can be returned by one method call; returning the \$list would force me to write much more XML to get its elements.

2 - Modify GetMainDB method implementation to do all mapping setup inside it. In this case I'm apparently don't need <Namespace> ... </Namespace> code fragment as it's nothing to do with the namespace besides adding global mapping.

[#API](#) [#Caché](#) [#Deployment](#) [#Development Environment](#) [#Mapping](#)

---

#### Source

URL:<https://community.intersystems.com/post/installer-manifest-how-partially-reconfigure-namespace-add-some-subscript-level-mapping>