Question
Robert Cemper · Nov 23, 2017

Where is the equivalent of INSPECTOR in Atelier ? See !

You probably have a similar experience.
The more I use Atelier the more I miss INSPECTOR that I had in Studio!

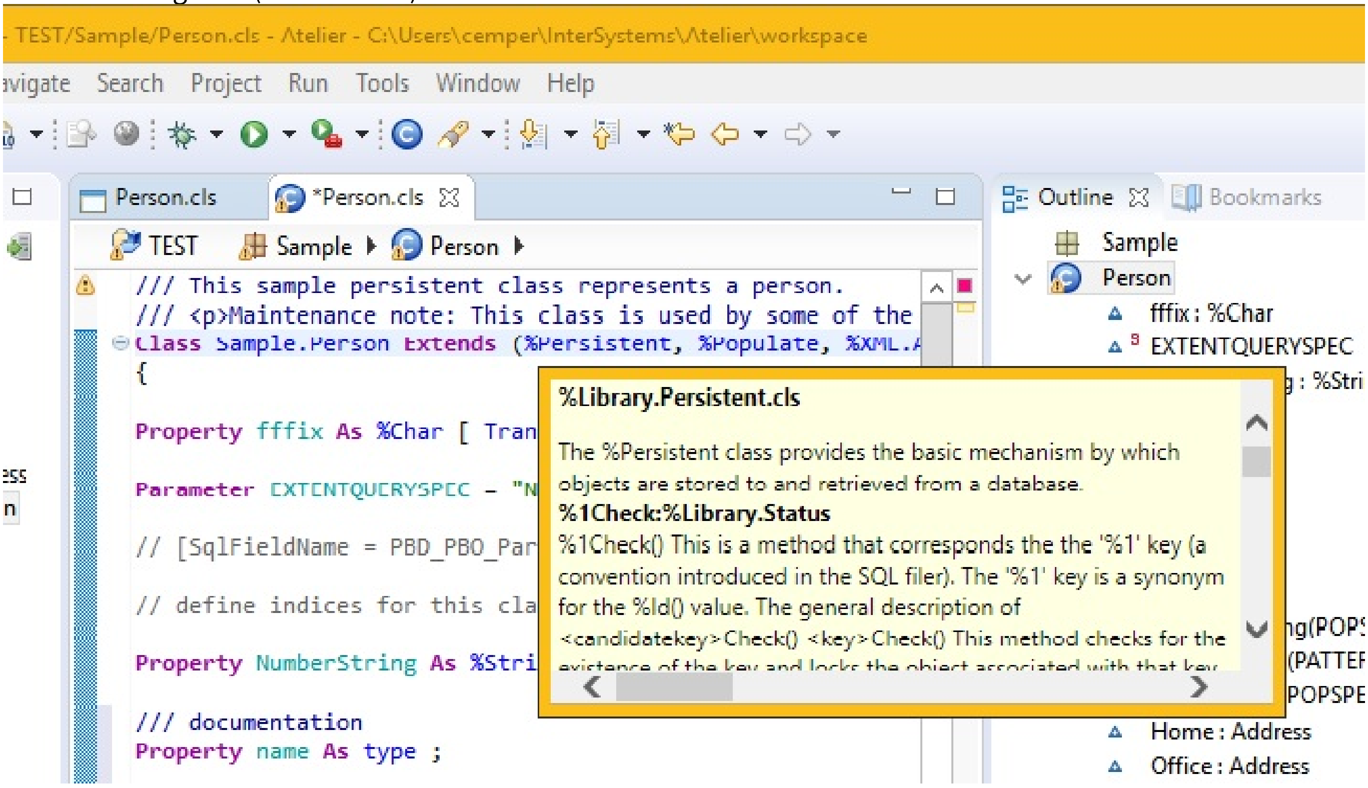
First for PROPERTIES that can inherit a variable number of Parameters beyond imagination.
take %XML.Adapter as the most obvious.
next:
Where to find all inherited methods that I eventually want to overload? Take %Persistent as example.

Or where or how can %OnBeforeAddToSaveSet() (the correct camel case) be overloaded ?

Or PARAMETERS:
Are we expected to know by heart every parameter around XML and SOAP?

It just can't be serious. There are a few relates links:
https://community.intersystems.com/post/atelier-and-inspector
https://community.intersystems.com/post/studio-view-inspector-atelier
Something similar is partially promised for 1.1 and later..

Further investigation (in Atelier 1.1) shows:



```
/// This sample persistent class represents a person.
/// <p>Maintenance note: This class is used by some of the
Class Sample.Person Extends (%Persistent, %Populate, %XML.A
{

Property fffix As %Char [ Tran

Parameter EXTENTQUERYSPEC = "N

// [SqlFieldName = PBD_PBO_Par

// define indices for this cla

Property NumberString As %Stri

/// documentation
Property name As type ;
```

%Library.Persistent.cls

The %Persistent class provides the basic mechanism by which objects are stored to and retrieved from a database.
%1Check:%Library.Status
%1Check() This is a method that corresponds the the '%1' key (a convention introduced in the SQL filer). The '%1' key is a synonym for the %Id() value. The general description of <candidatekey>Check() <key>Check() This method checks for the existence of the key and locks the object associated with that key

Moving the cursor on an EXPLICIT inherited class + <F2>
shows documentation on the explicit defined methods of this class.
nothing of the inherited ones!

In this example I see all methods defined in %Library.Persistent.

But still nothing inherited
from %Library.SwizzleObject, %Library.RegisteredObject, %Library.Base, %Library.SystemBase

The same applies to ClassPARAMETERS.
Those defined in the class are visible, nothing of inherited class parameters

Next I got an important advice from Andreas.Dieckow:
Hello Robert,

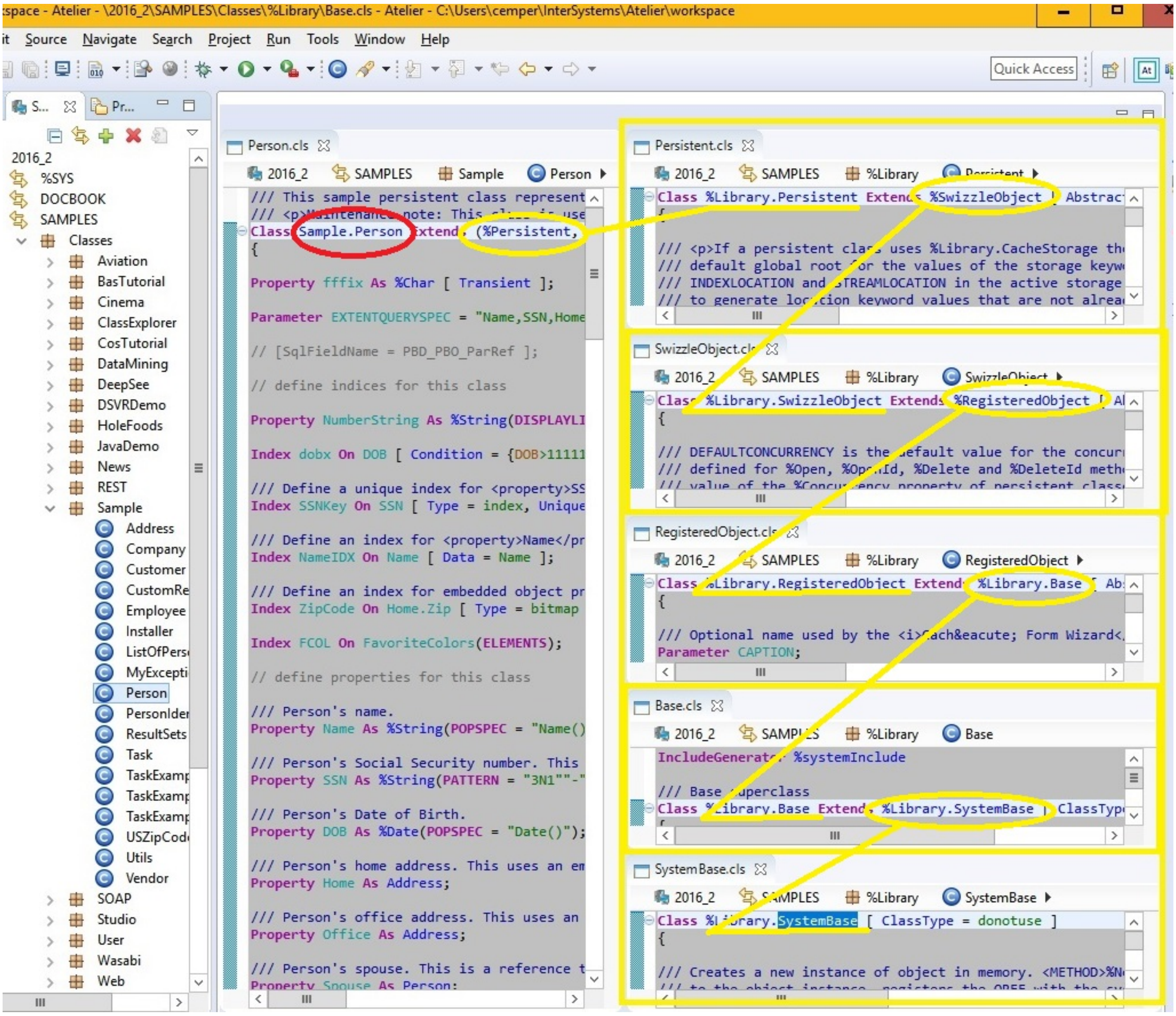
Right-click on a property, method etc. and choose “ Open declaration ” .
If the file the that gets opened is grayed in the background it means you cannot edit it,
because it is not part of the project.

I assume that is what you really are interested in.
Which I do not believe is what the inspector did in Studio.

Atelier also supports information on hovering and code completion as you type.
Also try Control-space for adding new code…

It's obvious that inherited classes just reside on server not on the local repository!
That's the core of the **paradigm change from Server based editing to Local editing**.
While in Studio everything is on server and local doesn't exist.

And so you can find all inherited classes immediately in Server Explorer and not in Atelier Explorer or Project Explorer
this is the way it looks if you do it right and open ALL inherited classes by click to **<F3>**



Person.cls
2016_2 SAMPLES Sample Person

```
/// This sample persistent class represent
/// <p>Maintenance note: This class is use
Class Sample.Person Extends (%Persistent,
{

Property fffix As %Char [ Transient ];

Parameter EXTENTQUERYSPEC = "Name,SSN,Home

// [SqlFieldName = PBD_PBO_ParRef ];

// define indices for this class

Property NumberString As %String(DISPLAYLI

Index dobx On DOB [ Condition = {DOB>11111

/// Define a unique index for <property>SS
Index SSNKey On SSN [ Type = index, Unique

/// Define an index for <property>Name</pr
Index NameIDX On Name [ Data = Name ];

/// Define an index for embedded object pr
Index ZipCode On Home.Zip [ Type = bitmap

Index FCOL On FavoriteColors(ELEMENTS);

// define properties for this class

/// Person's name.
Property Name As %String(POPSPEC = "Name()

/// Person's Social Security number. This
Property SSN As %String(PATTERN = "3N1""-"

/// Person's Date of Birth.
Property DOB As %Date(POPSPEC = "Date()");

/// Person's home address. This uses an em
Property Home As Address;

/// Person's office address. This uses an
Property Office As Address;

/// Person's spouse. This is a reference t
Property Spouse As Person;
```

Persistent.cls
2016_2 SAMPLES %Library Persistent

```
Class %Library.Persistent Extends %SwizzleObject [ Abstrac
{

/// <p>If a persistent class uses %Library.CacheStorage th
/// default global root for the values of the storage keyw
/// INDEXLOCATION and STREAMLOCATION in the active storage
/// to generate location keyword values that are not alrea
```

SwizzleObject.cls
2016_2 SAMPLES %Library SwizzleObject

```
Class %Library.SwizzleObject Extends %RegisteredObject [ Al
{

/// DEFAULTCONCURRENCY is the default value for the concur
/// defined for %Open, %OpenId, %Delete and %DeleteId meth
/// value of the %Concurrency property of persistent class
```

RegisteredObject.cls
2016_2 SAMPLES %Library RegisteredObject

```
Class %Library.RegisteredObject Extends %Library.Base [ Ab
{

/// Optional name used by the <i>Cach&eacute; Form Wizard<
Parameter CAPTION;
```

Base.cls
2016_2 SAMPLES %Library Base

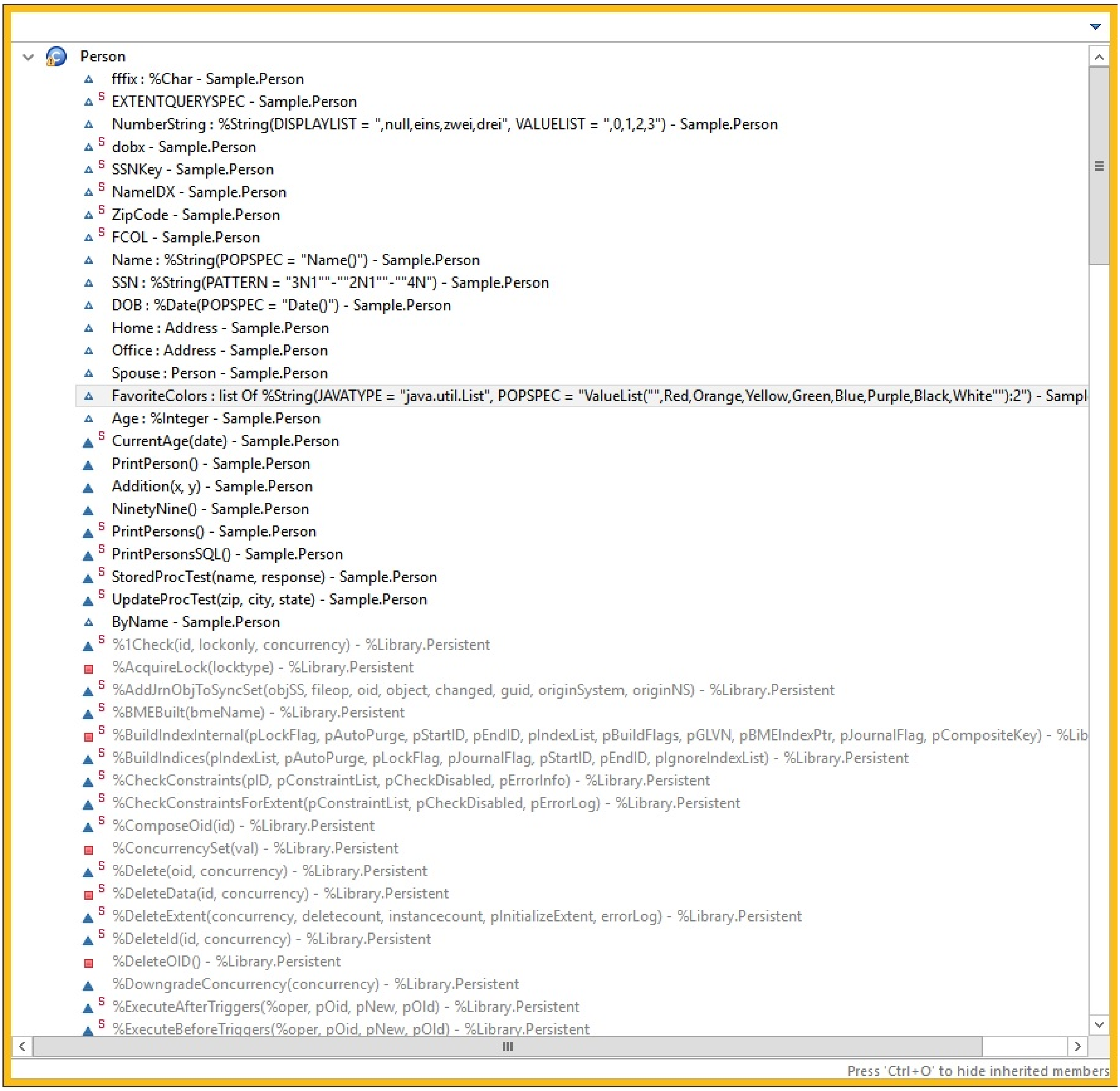
```
IncludeGenerator %systemInclude

/// Base superclass
Class %Library.Base Extends %Library.SystemBase [ ClassTyp
{
```

SystemBase.cls
2016_2 SAMPLES %Library SystemBase

```
Class %Library.SystemBase [ ClassType = donotuse ]
{

/// Creates a new instance of object in memory. <METHOD>%N
/// to the object instance, registers the OREF with the sy
```

The real **SOLUTION** was found by accident

put the cursor into the Editor of your class.
press **CTRL+O** . Quick Outline shows up.
press another **CTRL+O** and ALL inherited members of your class show up.



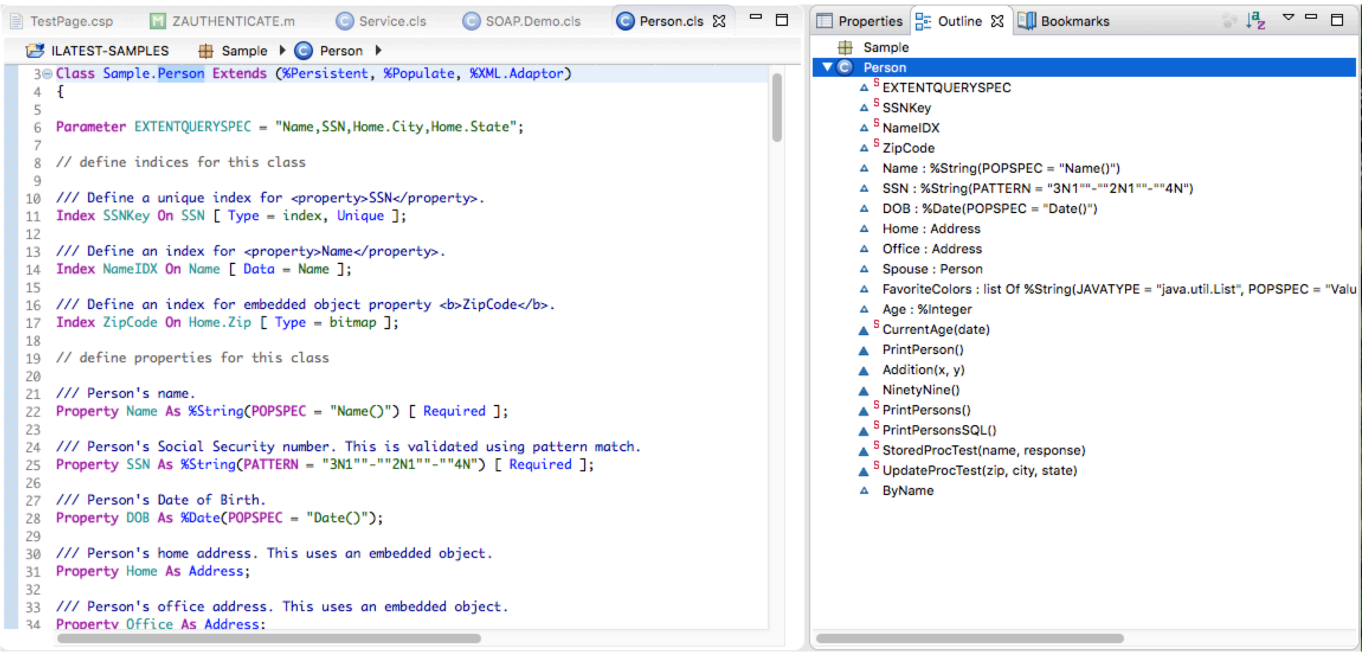
```
Person
  fffix : %Char - Sample.Person
S EXTENTQUERYSPEC - Sample.Person
  NumberString : %String(DISPLAYLIST = ",null,eins,zwei,drei", VALUELIST = ",0,1,2,3") - Sample.Person
S dobx - Sample.Person
S SSNKey - Sample.Person
S NameIDX - Sample.Person
S ZipCode - Sample.Person
S FCOL - Sample.Person
  Name : %String(POPSPEC = "Name()") - Sample.Person
  SSN : %String(PATTERN = "3N1""-""2N1""-""4N") - Sample.Person
  DOB : %Date(POPSPEC = "Date()") - Sample.Person
  Home : Address - Sample.Person
  Office : Address - Sample.Person
  Spouse : Person - Sample.Person
  FavoriteColors : list Of %String(JAVATYPE = "java.util.List", POPSPEC = "ValueList("",Red,Orange,Yellow,Green,Blue,Purple,Black,White""):2") - Sampl
  Age : %Integer - Sample.Person
S CurrentAge(date) - Sample.Person
  PrintPerson() - Sample.Person
  Addition(x, y) - Sample.Person
  NinetyNine() - Sample.Person
S PrintPersons() - Sample.Person
S PrintPersonsSQL() - Sample.Person
S StoredProcTest(name, response) - Sample.Person
S UpdateProcTest(zip, city, state) - Sample.Person
  ByName - Sample.Person
S %1Check(id, lockonly, concurrency) - %Library.Persistent
  %AcquireLock(locktype) - %Library.Persistent
S %AddJrnObjToSyncSet(objSS, fileop, oid, object, changed, guid, originSystem, originNS) - %Library.Persistent
S %BMEBuilt(bmeName) - %Library.Persistent
S %BuildIndexInternal(pLockFlag, pAutoPurge, pStartID, pEndID, pIndexList, pBuildFlags, pGLVN, pBMEIndexPtr, pJournalFlag, pCompositeKey) - %Lib
S %BuildIndices(pIndexList, pAutoPurge, pLockFlag, pJournalFlag, pStartID, pEndID, pIgnoreIndexList) - %Library.Persistent
S %CheckConstraints(pID, pConstraintList, pCheckDisabled, pErrorInfo) - %Library.Persistent
S %CheckConstraintsForExtent(pConstraintList, pCheckDisabled, pErrorLog) - %Library.Persistent
S %ComposeOid(id) - %Library.Persistent
  %ConcurrencySet(val) - %Library.Persistent
S %Delete(oid, concurrency) - %Library.Persistent
S %DeleteData(id, concurrency) - %Library.Persistent
S %DeleteExtent(concurrency, deletecount, instancecount, pInitializeExtent, errorLog) - %Library.Persistent
S %DeleteId(id, concurrency) - %Library.Persistent
  %DeleteOID() - %Library.Persistent
  %DowngradeConcurrency(concurrency) - %Library.Persistent
S %ExecuteAfterTriggers(%oper, pOid, pNew, pOld) - %Library.Persistent
S %ExecuteBeforeTriggers(%oper, pOid, pNew, pOld) - %Library.Persistent
```

Press 'Ctrl+O' to hide inherited members

Almost the same time I received a confirming answer from Nicole Aaron

++

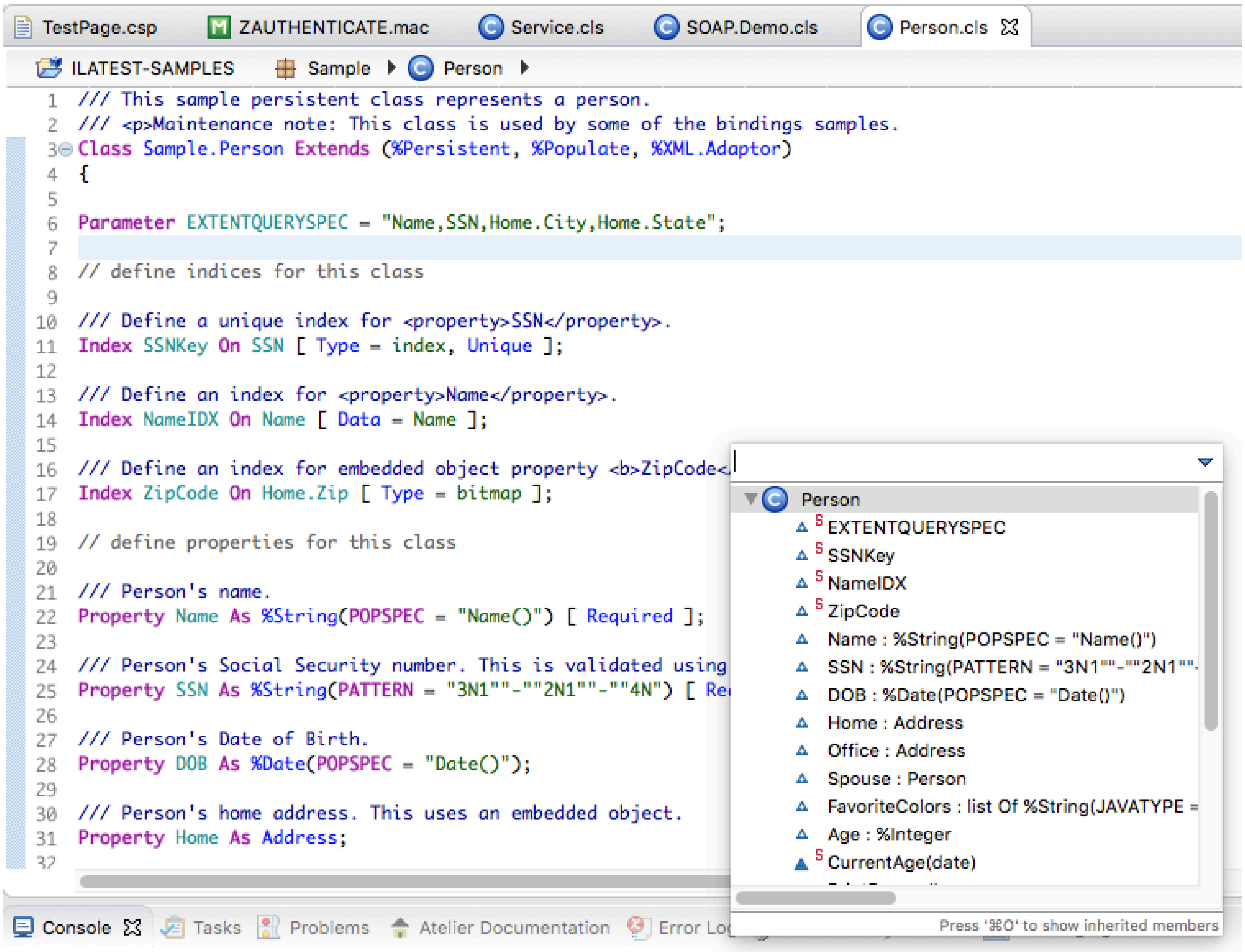
I just wanted to share a couple other Atelier tools I find useful that have some similar functionality to the Studio Inspector. The first is the Atelier Outline View. This shows the currently defined members of a file (parameters, properties, methods, etc):



```
Class Sample.Person Extends (%Persistent, %Populate, %XML.Adaptor)
{

Parameter EXTENTQUERYSPEC = "Name,SSN,Home.City,Home.State";

// define indices for this class

/// Define a unique index for <property>SSN</property>.
Index SSNKey On SSN [ Type = index, Unique ];

/// Define an index for <property>Name</property>.
Index NameIDX On Name [ Data = Name ];

/// Define an index for embedded object property <b>ZipCode</b>.
Index ZipCode On Home.Zip [ Type = bitmap ];

// define properties for this class

/// Person's name.
Property Name As %String(POPSPEC = "Name()") [ Required ];

/// Person's Social Security number. This is validated using pattern match.
Property SSN As %String(PATTERN = "3N1""-""2N1""-""4N") [ Required ];

/// Person's Date of Birth.
Property DOB As %Date(POPSPEC = "Date()");

/// Person's home address. This uses an embedded object.
Property Home As Address;

/// Person's office address. This uses an embedded object.
Property Office As Address;
```

```
Sample
Person
    S EXTENTQUERYSPEC
    S SSNKey
    S NameIDX
    S ZipCode
      Name : %String(POPSPEC = "Name()")
      SSN : %String(PATTERN = "3N1""-""2N1""-""4N")
      DOB : %Date(POPSPEC = "Date()")
      Home : Address
      Office : Address
      Spouse : Person
      FavoriteColors : list Of %String(JAVATYPE = "java.util.List", POPSPEC = "Valu
      Age : %Integer
    S CurrentAge(date)
      PrintPerson()
      Addition(x, y)
      NinetyNine()
    S PrintPersons()
    S PrintPersonsSQL()
    S StoredProcTest(name, response)
    S UpdateProcTest(zip, city, state)
      ByName
```

The second tool is the Quick Outline, which you access using cmd+O on Mac or ctrl+O on Windows. The first time you hit cmd+O/ctrl+O you will see a list of members defined in the current class:

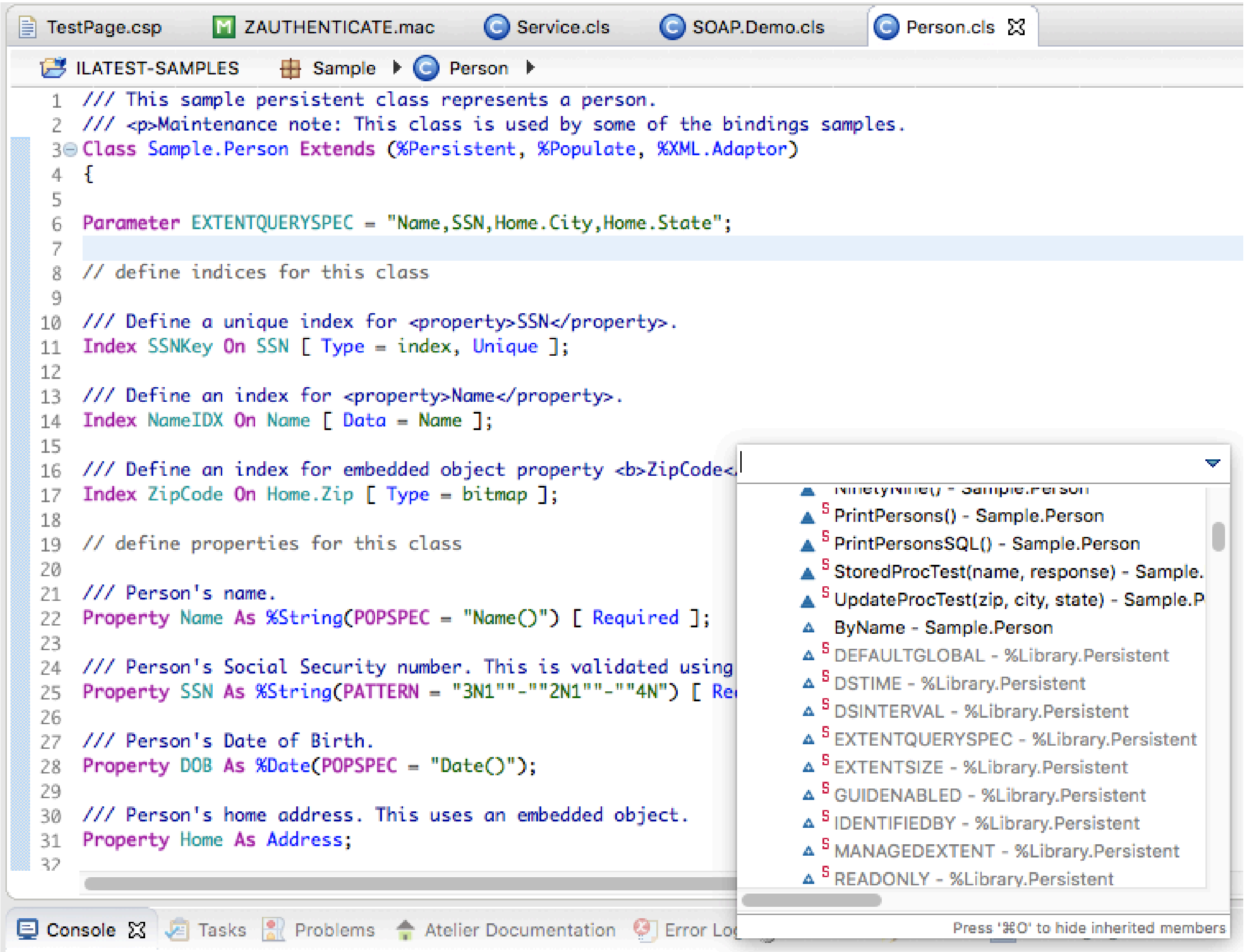


```
/// This sample persistent class represents a person.
/// <p>Maintenance note: This class is used by some of the bindings samples.
Class Sample.Person Extends (%Persistent, %Populate, %XML.Adaptor)
{

Parameter EXTENTQUERYSPEC = "Name,SSN,Home.City,Home.State";

// define indices for this class

/// Define a unique index for <property>SSN</property>.
Index SSNKey On SSN [ Type = index, Unique ];

/// Define an index for <property>Name</property>.
Index NameIDX On Name [ Data = Name ];

/// Define an index for embedded object property <b>ZipCode<
Index ZipCode On Home.Zip [ Type = bitmap ];

// define properties for this class

/// Person's name.
Property Name As %String(POPSPEC = "Name()") [ Required ];

/// Person's Social Security number. This is validated using
Property SSN As %String(PATTERN = "3N1""-""2N1""-""4N") [ Re

/// Person's Date of Birth.
Property DOB As %Date(POPSPEC = "Date()");

/// Person's home address. This uses an embedded object.
Property Home As Address;
```

```
Person
    S EXTENTQUERYSPEC
    S SSNKey
    S NameIDX
    S ZipCode
      Name : %String(POPSPEC = "Name()")
      SSN : %String(PATTERN = "3N1""-""2N1""-
      DOB : %Date(POPSPEC = "Date()")
      Home : Address
      Office : Address
      Spouse : Person
      FavoriteColors : list Of %String(JAVATYPE =
      Age : %Integer
    S CurrentAge(date)
```

Press '⌘O' to show inherited members

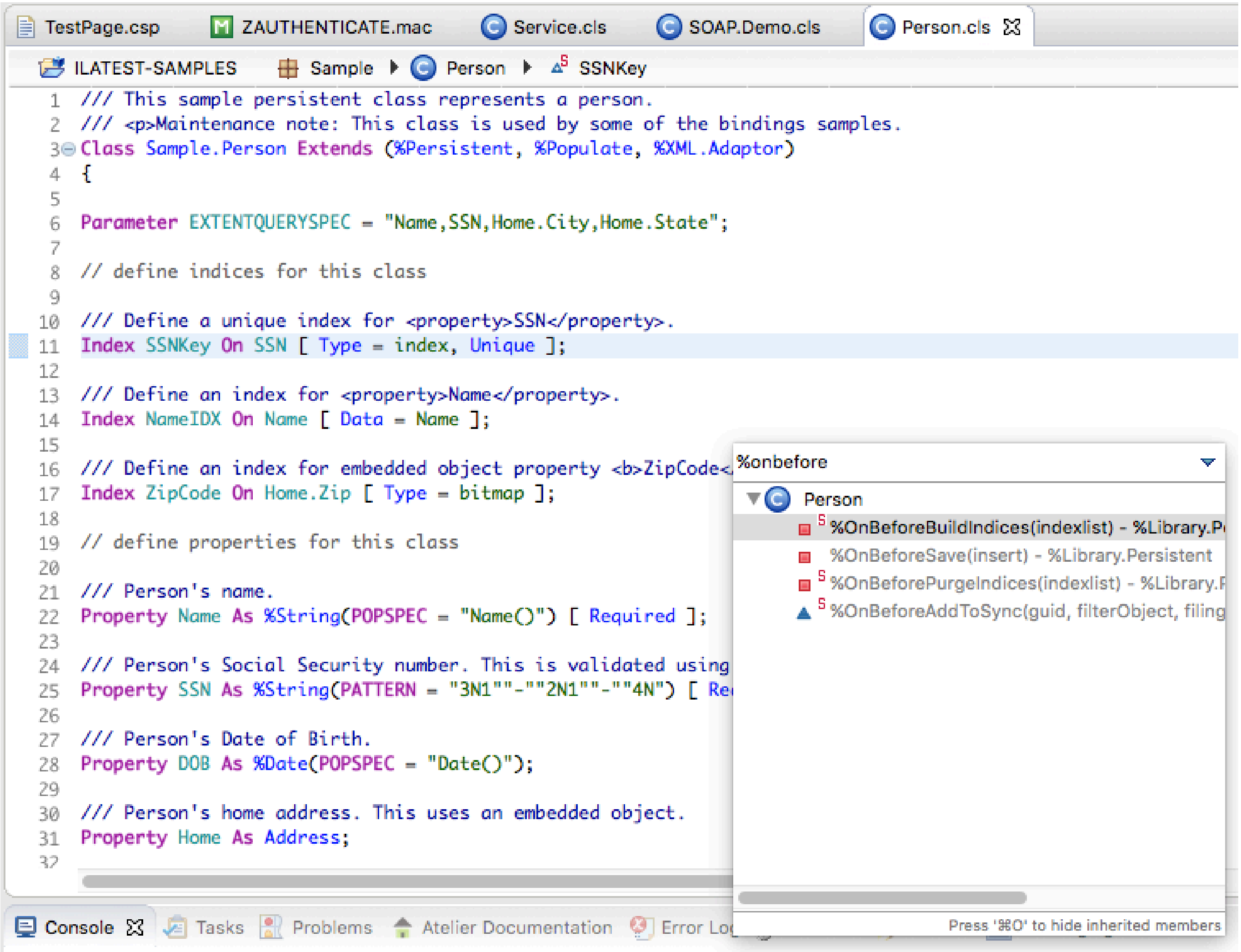
You can then hit cmd+O/ctrl+O again to show the inherited members, along with which class each member comes from. The inherited members show up in a lighter gray font instead of black:



```
/// This sample persistent class represents a person.
/// <p>Maintenance note: This class is used by some of the bindings samples.
Class Sample.Person Extends (%Persistent, %Populate, %XML.Adaptor)
{

Parameter EXTENTQUERYSPEC = "Name,SSN,Home.City,Home.State";

// define indices for this class

/// Define a unique index for <property>SSN</property>.
Index SSNKey On SSN [ Type = index, Unique ];

/// Define an index for <property>Name</property>.
Index NameIDX On Name [ Data = Name ];

/// Define an index for embedded object property <b>ZipCode</
Index ZipCode On Home.Zip [ Type = bitmap ];

// define properties for this class

/// Person's name.
Property Name As %String(POPSPEC = "Name()") [ Required ];

/// Person's Social Security number. This is validated using
Property SSN As %String(PATTERN = "3N1""-""2N1""-""4N") [ Re

/// Person's Date of Birth.
Property DOB As %Date(POPSPEC = "Date()");

/// Person's home address. This uses an embedded object.
Property Home As Address;
```

```
NinetyNine() - Sample.Person
PrintPersons() - Sample.Person
PrintPersonsSQL() - Sample.Person
StoredProcTest(name, response) - Sample.
UpdateProcTest(zip, city, state) - Sample.P
ByName - Sample.Person
DEFAULTGLOBAL - %Library.Persistent
DSTIME - %Library.Persistent
DSINTERVAL - %Library.Persistent
EXTENTQUERYSPEC - %Library.Persistent
EXTENTSIZE - %Library.Persistent
GUIDENABLED - %Library.Persistent
IDENTIFIEDBY - %Library.Persistent
MANAGEDEXTENT - %Library.Persistent
READONLY - %Library.Persistent
```

You can also start typing in this box to narrow down the list:


TestPage.csp
ZAUTHENTICATE.mac
Service.cls
SOAP.Demo.cls
Person.cls
ILATEST-SAMPLES
Sample
Person
SSNKey
%onbefore
Person
%OnBeforeBuildIndices(indexlist) - %Library.P
%OnBeforeSave(insert) - %Library.Persistent
%OnBeforePurgeIndices(indexlist) - %Library.F
%OnBeforeAddToSync(guid, filterObject, filing
Press '⌘O' to hide inherited members
Console
Tasks
Problems
Atelier Documentation
Error Lo

```
/// This sample persistent class represents a person.
/// <p>Maintenance note: This class is used by some of the bindings samples.
Class Sample.Person Extends (%Persistent, %Populate, %XML.Adaptor)
{

Parameter EXTENTQUERYSPEC = "Name,SSN,Home.City,Home.State";

// define indices for this class

/// Define a unique index for <property>SSN</property>.
Index SSNKey On SSN [ Type = index, Unique ];

/// Define an index for <property>Name</property>.
Index NameIDX On Name [ Data = Name ];

/// Define an index for embedded object property <b>ZipCode<
Index ZipCode On Home.Zip [ Type = bitmap ];

// define properties for this class

/// Person's name.
Property Name As %String(POPSPEC = "Name()") [ Required ];

/// Person's Social Security number. This is validated using
Property SSN As %String(PATTERN = "3N1""-""2N1""-""4N") [ Re

/// Person's Date of Birth.
Property DOB As %Date(POPSPEC = "Date()");

/// Person's home address. This uses an embedded object.
Property Home As Address;
```

#Beginner #Object Data Model #InterSystems IRIS

Source URL:https://community.intersystems.com/post/where-equivalent-inspector-atelier-see