GEORGEJAMES
SOFTWARE

Yuzinji

InterSystems Objects flavor for Structure101g

*Yuzinji Installation and
Operation Guide*

Product Support

Support is available to all users of George James Software Products who have a current Software Maintenance Agreement. Support and assistance can be obtained from the following sources:

Telephone   +44-1932-252568
E-mail   support@georgejames.com
Web page   www.georgejames.com/support

Rev	Document Reference	Date	Prepared by	Details
0	C:\Users\johnm.GJ\Desktop\Yuzinji_Installation_Operation.0.doc	23 Jul 2013	John Murray	First release (Yuzinji 1.0.1)
1	\\flo\wiz\Lib\Product_Manuals\Yuzinji\Yuzinji_Installation_Operation.1.doc	16 Aug 2013	John Murray	Yuzinji 1.0.2 (instructions for upgrading)
2	\\flo\wiz\Lib\Product_Manuals\Yuzinji\Yuzinji_Installation_Operation.2.doc	27 Sep 2013	John Murray	Yuzinji 1.0.3
3	\\flo\Products\Yuzinji\Manuals\Yuzinji_Installation_Operation.3.doc	9 Mar 2016	John Murray	Yuzinji 1.1.0
4	\\flo\Products\Yuzinji\Manuals\Yuzinji_Installation_Operation.4.doc	25 Sep 2017	John Murray	Yuzinji 1.1.2
5	\\flo\Products\Yuzinji\Manuals\Yuzinji_Installation_Operation.5.doc	29 May 2018	John Murray	Yuzinji 1.2.0 (support for InterSystems IRIS)

# Yuzinji Installation and Operation Guide

Contents

1   Installation

Yuzinji consists of two parts. A flavor (plug-in) extends Structure101g from Headway Software (www.structure101.com), and a server-side component enables ObjectScript code in an InterSystems IRIS, Caché or Ensemble server (www.intersystems.com) to be analyzed and visualized in Structure101g
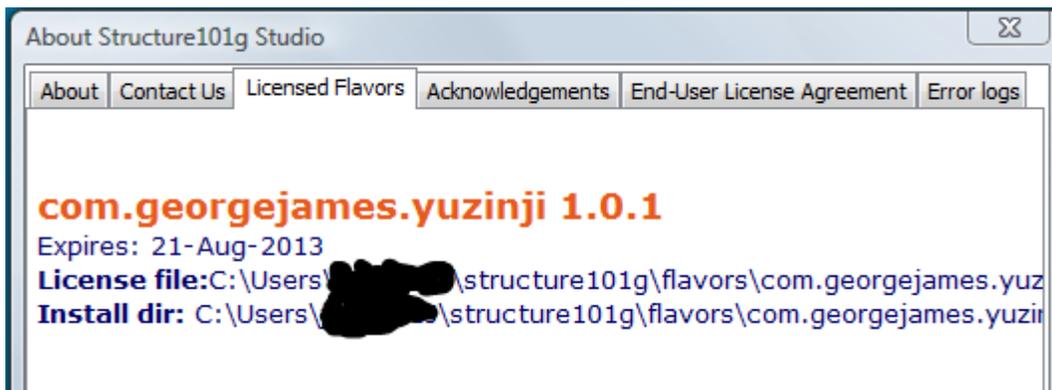
This section describes first-time installation. If you are updating an existing Yuzinji instance to a newer version please consult a later section of the document.

Flavor Installation

Using the Install option on the Flavors menu, add the Yuzinji flavor to Structure101g Studio 4.0 or later.
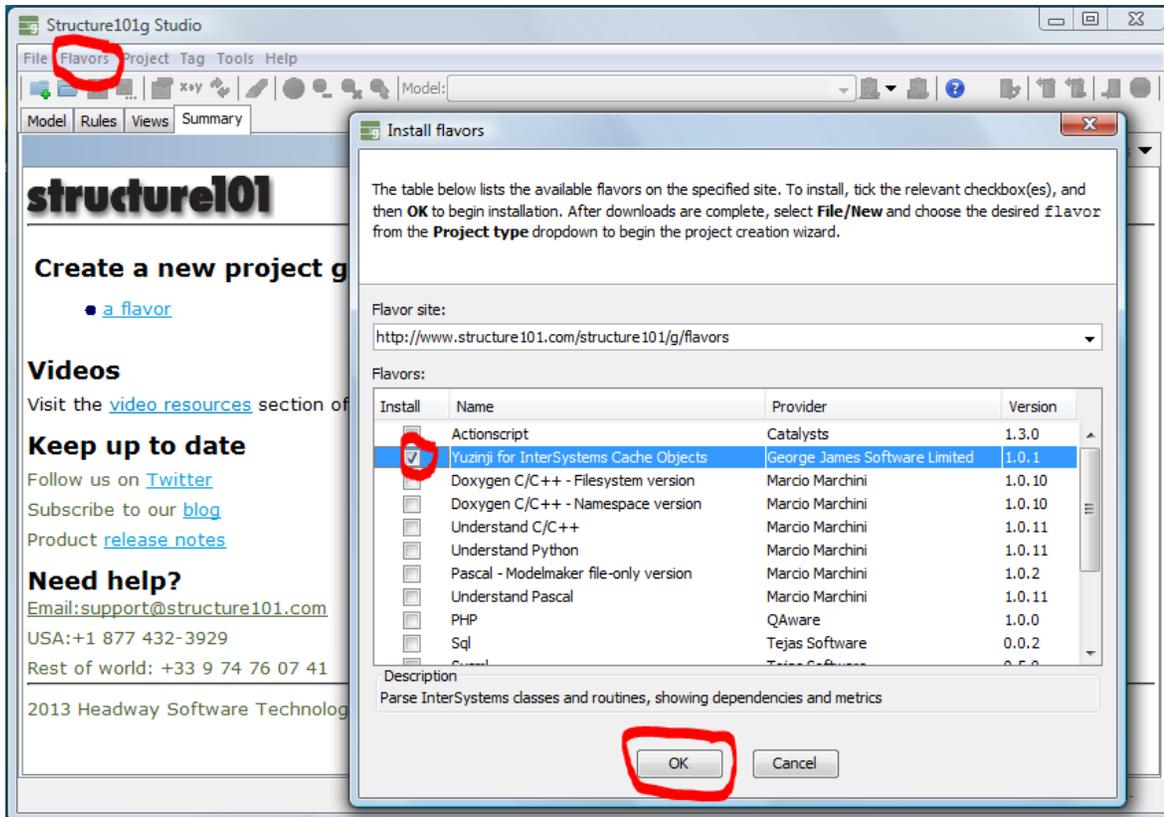
Use Help\About to view the Licensed Flavors tab:

The **Install dir** value is selectable and can be copied to the system clipboard using Ctrl+C or equivalent keystroke. Note that you may need to scroll right to see the entire directory path.

Installation provides a 30-day evaluation license for the flavor. Notice the expiry date and the path to the evaluation license file.

If a full license has been purchased for the flavor a license file will be issued. This file should be placed in the **flavors** directory of the path shown in the dialog, e.g. c:\users*username*\structure101g\flavors\. When Structure101g Studio is restarted, the **Licensed Flavors** tab of the About dialog will no longer show an expiry date.

Server-Side Component Installation

On the computer where the flavor was added to Structure101g Studio, browse to the **serverkits** subdirectory of the flavor install directory path that was noted in the previous step.

Choose the zipfile matching your server version. For instance:

    InterSystems2010.2plus-yuzinji1.2.0.zip

is suitable for InterSystems Caché and Caché-based platforms (e.g. Ensemble) version 2010.2 or greater, and for InterSystems IRIS.

Unzip this file, which gives you a **yuzinji** subdirectory.

Move this subdirectory into the **devuser\studio\templates** subdirectory of your installed InterSystems product.

If you unzipped on a Windows platform and are moving the subdirectory and its contents to a platform where textfiles use a different line terminator, make sure your transfer utility treats the CACHE.DAT or IRIS.DAT file as binary, and the other files as text. For example, if using FTP, use TYPE I or TYPE A respectively.

If the server platform is UNIX / Linux / macOS, recursively assign the owner:group of the resulting **devuser/studio/templates/yuzinji** directory and its contents to the owner:group that the server instance runs as, e.g. cacheusr:cacheusr or irisusr:irisusr

Start a Studio session. Connect as a user with sufficient privileges to amend the InterSystems platform's configuration.

> If Studio is not available to you, use Atelier instead. If Atelier's Add-Ins option doesn't give a way of refreshing the server's set of add-ins, run the following command in a terminal session in the %SYS namespace to achieve that:
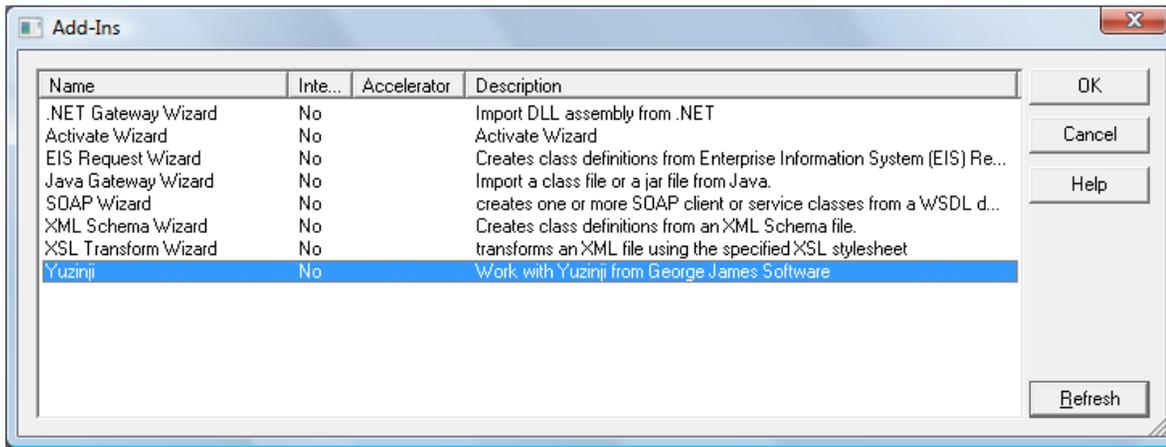>
> ```
> d ##class(%CSP.StudioTemplateMgr).UpdateTemplates()
> ```

Connect to the SAMPLES namespace (if you don't have SAMPLES, use USER or else one of your app development namespaces).
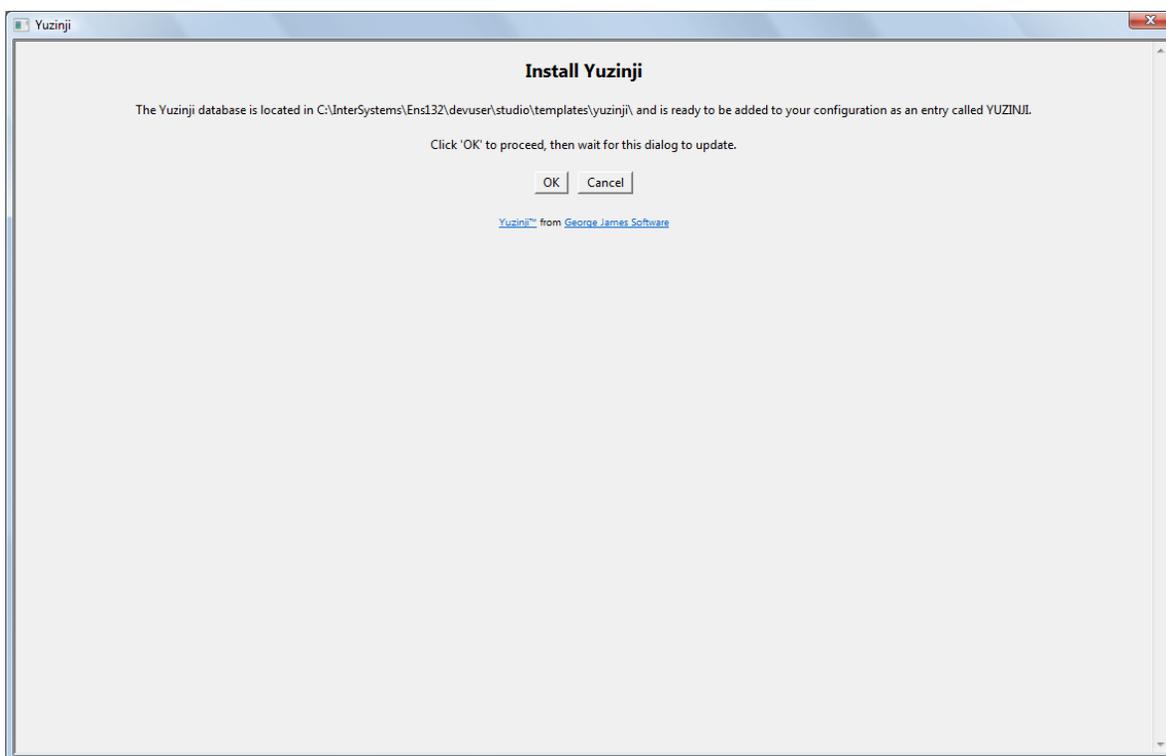
On the "Tools" menu choose "Add-Ins\Add-Ins..."
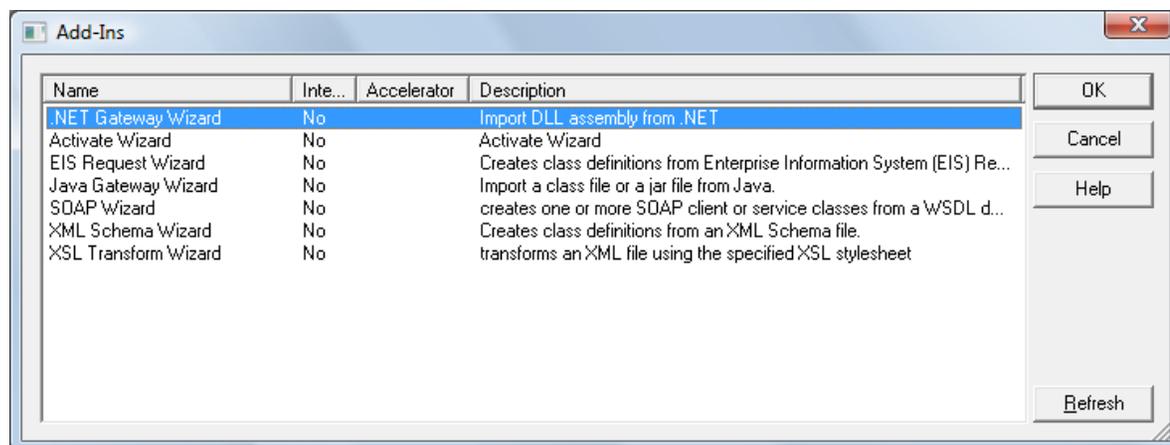
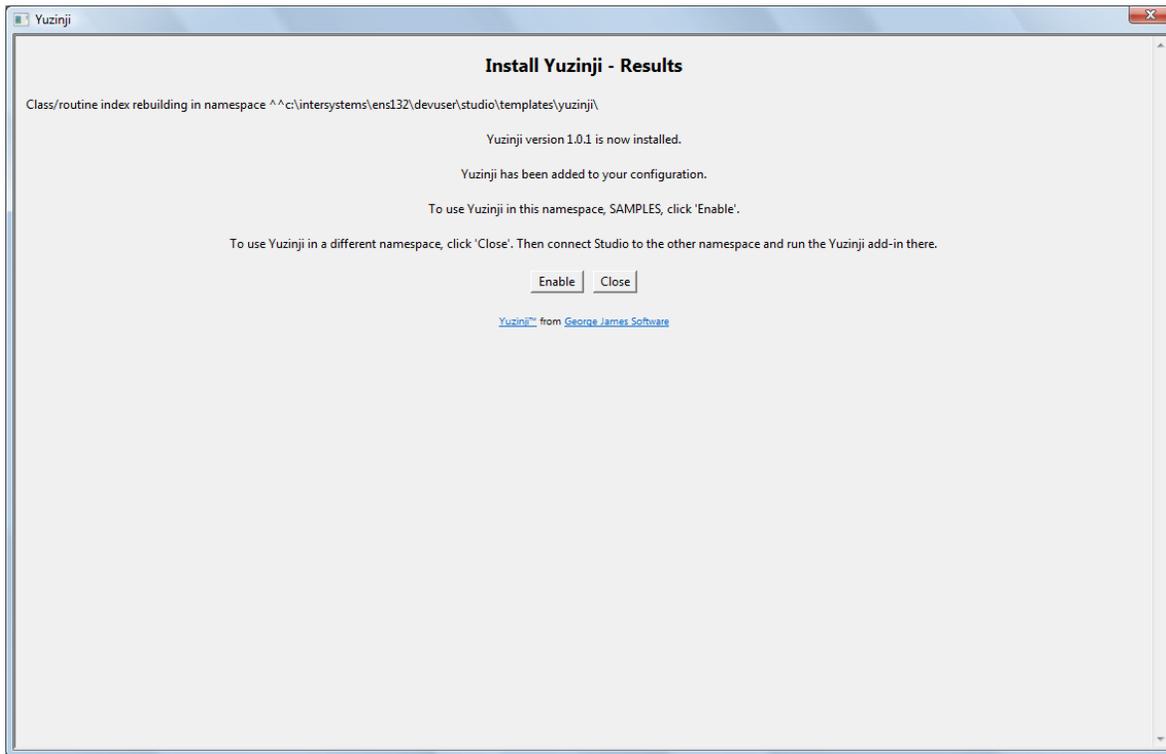Click the **Refresh** button on the dialog.

When the refresh has completed, choose the new Yuzinji add-in and click **OK**.

Confirm that you want to add the YUZINJI database:

Install Yuzinji

The Yuzinji database is located in C:\InterSystems\Ens132\devuser\studio\templates\yuzinji\ and is ready to be added to your configuration as an entry called YUZINJI.

Click 'OK' to proceed, then wait for this dialog to update.

OK   Cancel

Yuzinji™ from George James Software

When this completes the server-side component of Yuzinji is available for use:

Install Yuzinji - Results

Class/routine index rebuilding in namespace ^^c:\intersystems\ens132\devuser\studio\templates\yuzinji\

Yuzinji version 1.0.1 is now installed.

Yuzinji has been added to your configuration.

To use Yuzinji in this namespace, SAMPLES, click 'Enable'.

To use Yuzinji in a different namespace, click 'Close'. Then connect Studio to the other namespace and run the Yuzinji add-in there.

Enable   Close

Yuzinji™ from George James Software

You can exit at this point using the **Close** button, or you may proceed to Yuzinji-enable the current namespace by clicking the **Enable** button.

2   First Use in a Namespace

When the Yuzinji add-in is run in a namespace for the first time you must grant permission for a partner namespace and database to be created, and for mappings to be added to the current namespace definition.
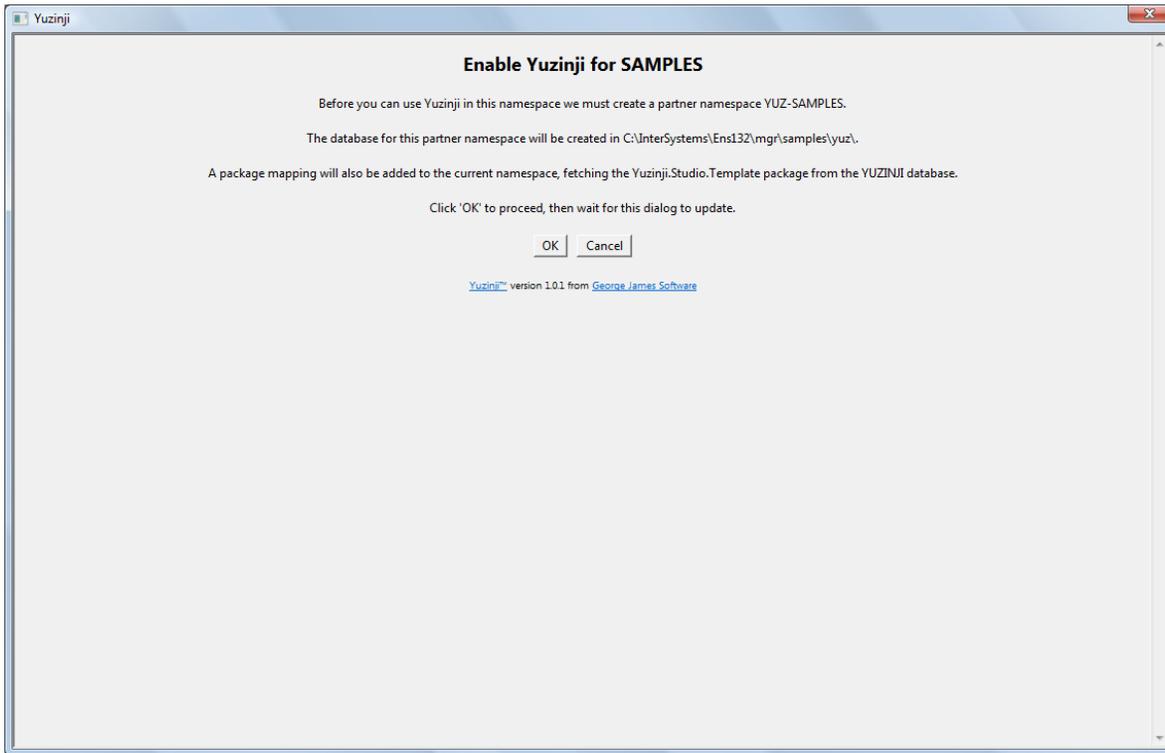
Your Studio or Atelier connection must be made as a user with sufficient privilege to amend the configuration in this manner.
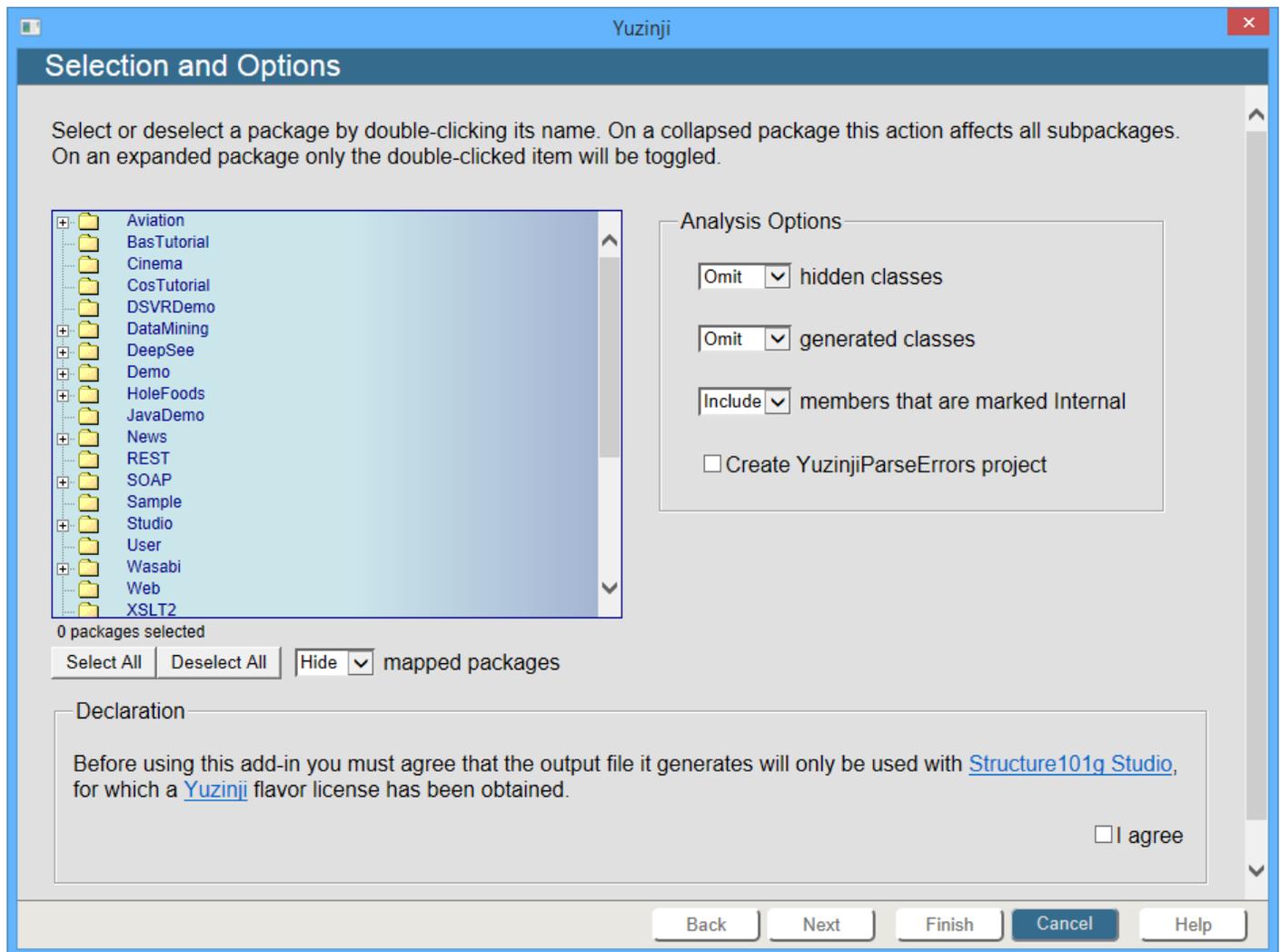
Click **OK** to proceed, then wait for the next page.
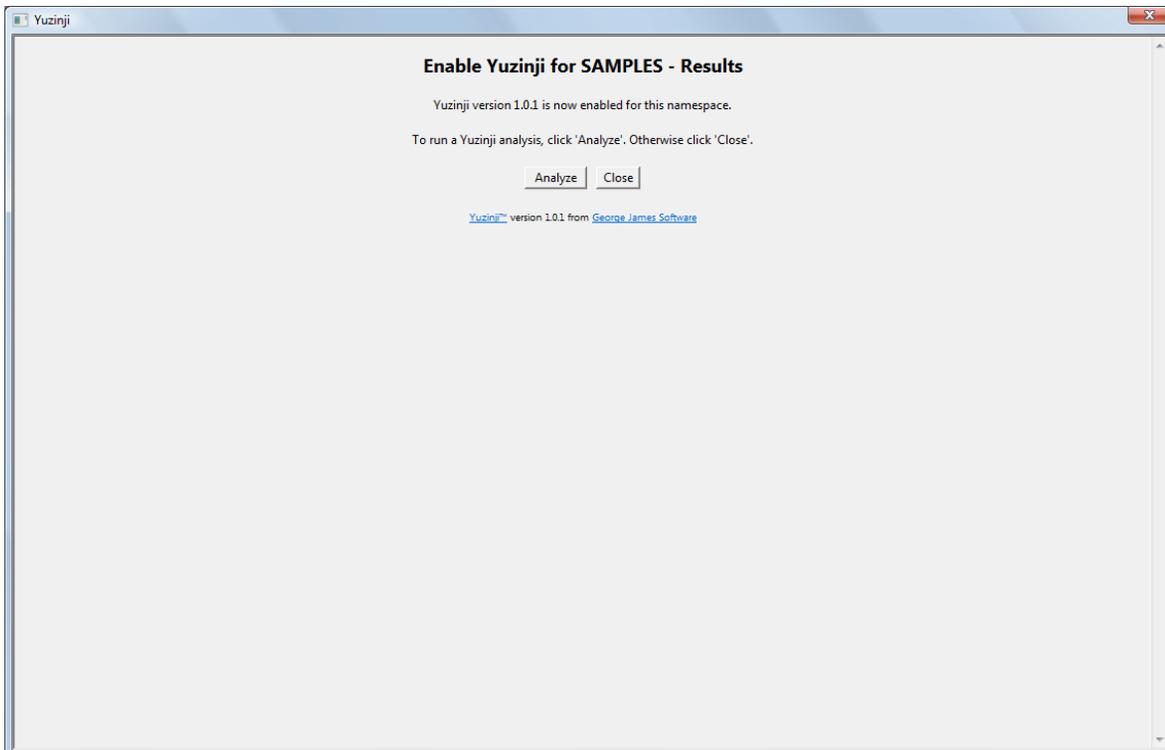
At this point you can **Close** the add-in, or you can proceed to **Analyze** some code in the namespace.
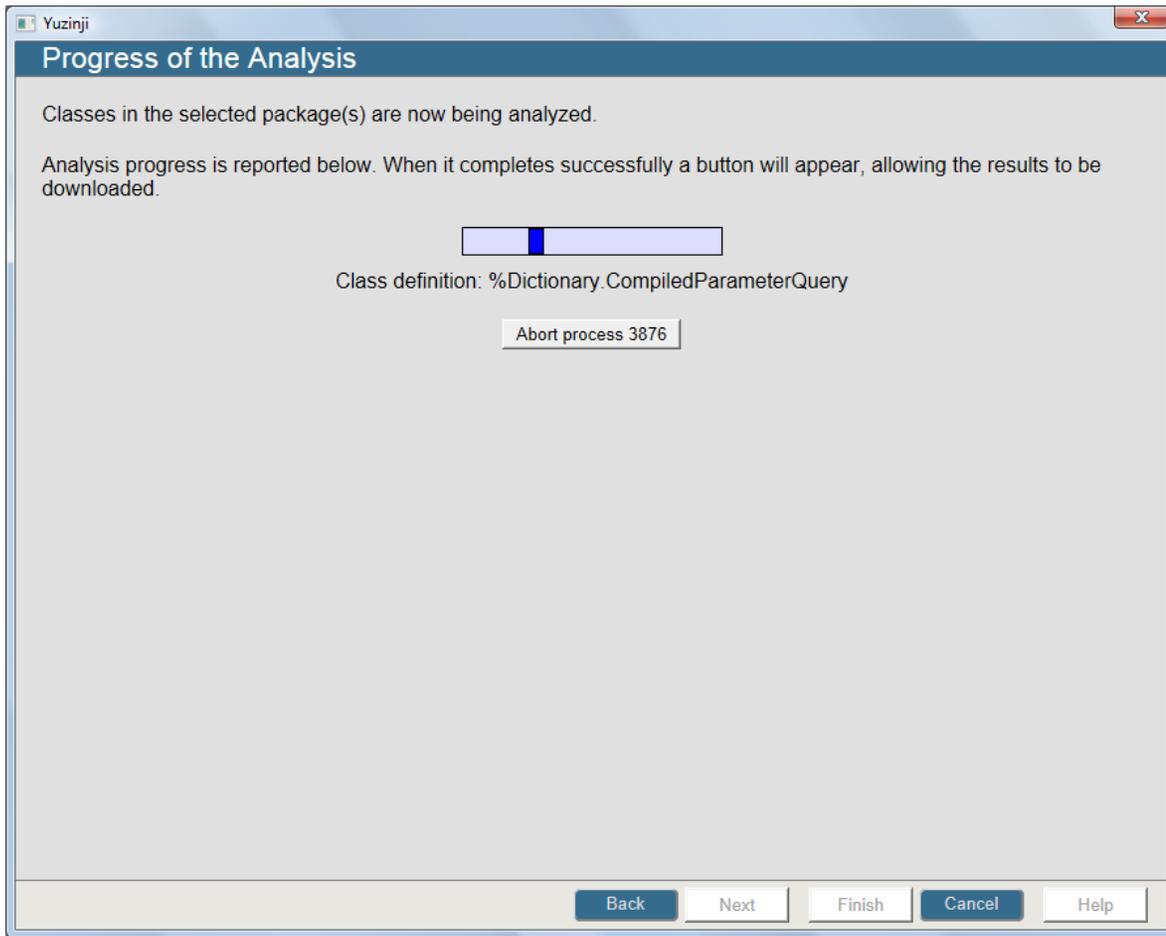
3   Analyzing Code

Whenever the Yuzinji add-in is run in a Yuzinji-enabled namespace you will be presented with the following dialog:
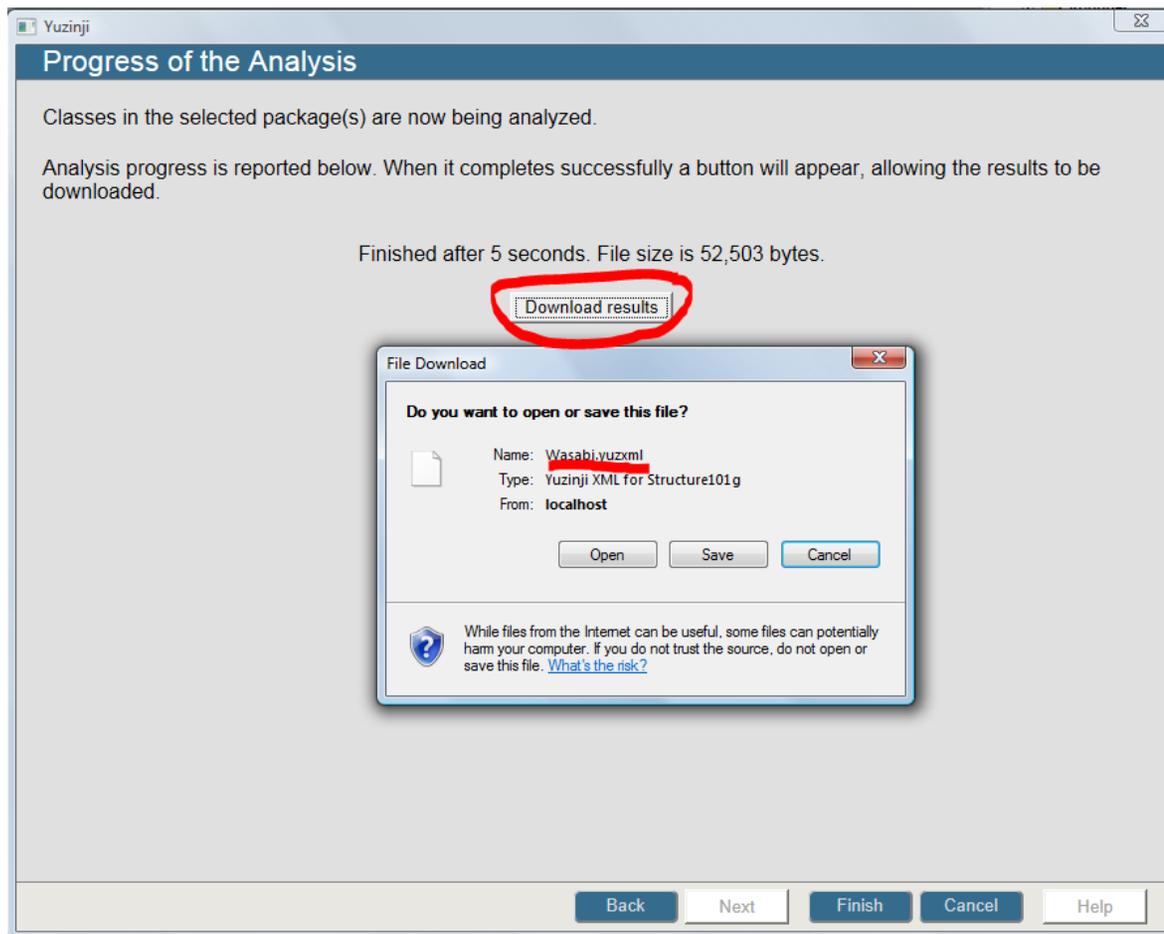

Select one of more packages (e.g. Wasabi in the SAMPLES namespace). To select or deselect a package, double-click on its name. If the package is collapsed this action affects all subpackages, but on an expanded package only the double-clicked item will be toggled. Note also the **Select All** and **Deselect All** buttons.

The Analysis Options panel gives you some control of what will be included in the output. It also lets you create a Studio project containing any classes or routines with code that Yuzinji failed to parse completely. That project is always named YuzinjiParseErrors, and will be deleted prior to the analysis run. If no errors are encountered then no project will be created.

On the first use of this dialog during each Studio session you must check the **I agree** box to confirm you agree to the license conditions. Then click **Next** to start the analysis.

Progress of the Analysis

Classes in the selected package(s) are now being analyzed.

Analysis progress is reported below. When it completes successfully a button will appear, allowing the results to be downloaded.

Class definition: %Dictionary.CompiledParameterQuery

Abort process 3876

Back   Next   Finish   Cancel   Help

When analysis completes, use the **Download results** button that appears:

Either save it or open it, setting the .yuzxml extension to run the Structure101g Studio executable.

If you opted to save the downloaded file rather than to run it immediately, you can subsequently open it from within Structure101g Studio by choosing the XML file type on the File\Open dialog.

4 Yuzinji and Structure101g Studio

4.1 Analyzing Routines

The Yuzinji add-in for Studio and Atelier only allows packages of classes to be selected. Wherever class method code invokes routines (e.g. with subroutine calls or extrinsic function calls) a stub item (see below) will be created for the routine. If you need to analyze routine code explicitly (MAC or INT), add the routines to a Studio project, save the project, then invoke the Yuzinji add-in from Studio while this project is open.
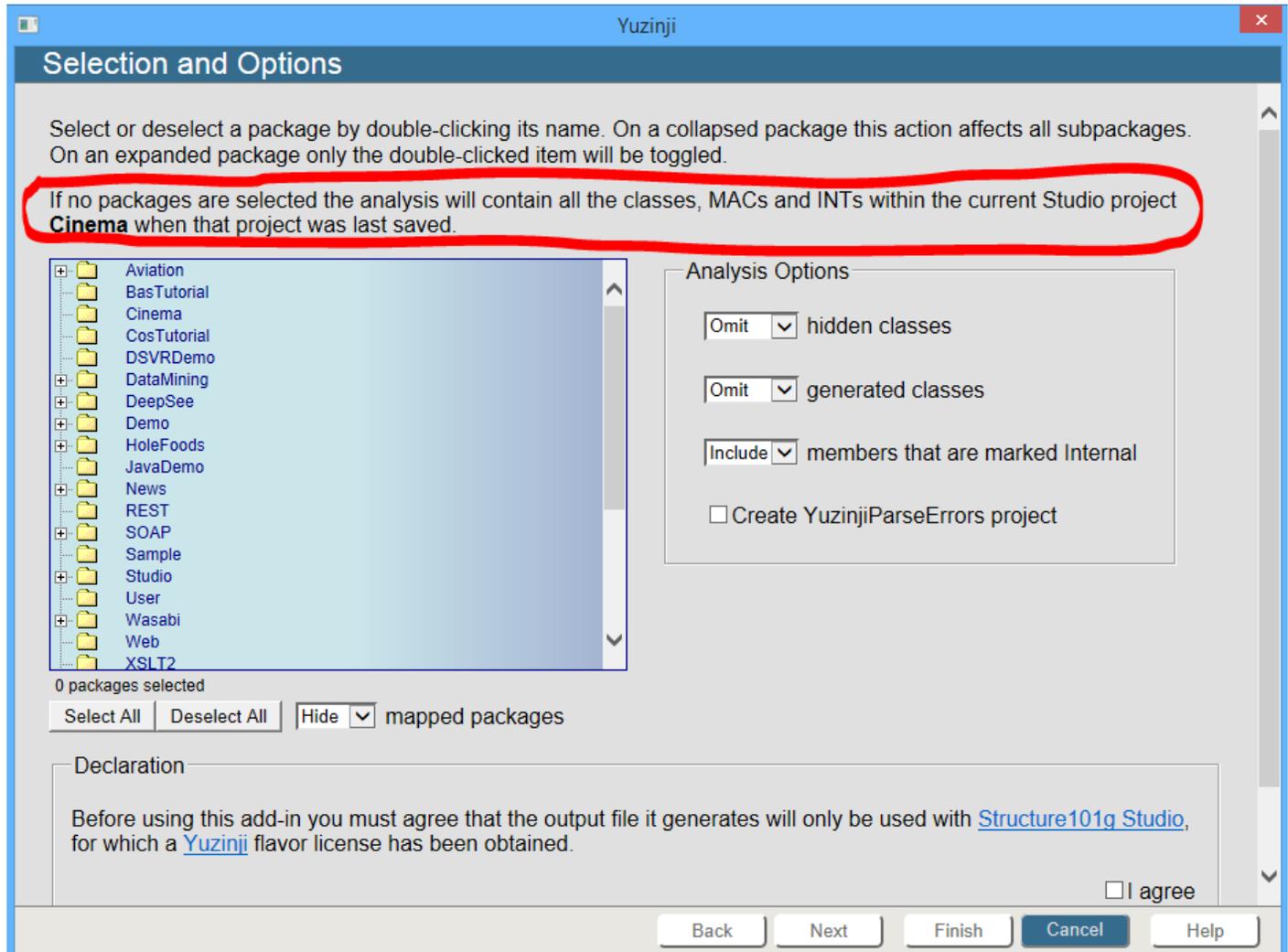
Observe the additional dialog text. Select no packages in the list, but continue to the next page. The result file will contain all the packages, classes, MACs and INTs that were on the project when last saved.

This technique is also convenient if you want to analyze the same set of packages repeatedly without having to select them within the list each time.

4.2 Result Caching

A class or routine that has previously been analyzed will only be reanalyzed if its "last modified" timestamp changes.

4.3 Analysis via API

A Yuzinji analysis can be run programmatically by using the $$project^reyuzapi entrypoint.

Here is the documentation:

```
project(prj,file,opts) ;;API; Analyze all CLS, MAC and INT members of Studio project prj, using options in
opts(). Write results to file.
```

```
    ;; INPUTS:
    ;; prj => Name of Studio project whose PKG, CLS, MAC and INT members are to be analyzed.
    ;;        If this API function is called in a YUZ-XYZ namespace, seek the project and its members in the
XYZ namespace
    ;; file => Name of file to contain the analysis results (.yuzxml extension recommended)
    ;;
    ;; opts("excludeGenerated")=1 => Do not analyze classes that are marked as GeneratedBy; default=0
    ;; opts("excludeInternal")=1 => Do not show members (e.g. methods) that are marked Internal (these do not
appear in online class doc); default=0
    ;; opts("excludeRel","parent")=1 => Relationship property of type 'parent' does not constitute a
dependency; default=0
    ;; opts("excludeRel","children")=1 => Relationship property of type 'children' does not constitute a
dependency; default=1
    ;; opts("excludeRel","one")=1 => Relationship property of type 'one' does not constitute a dependency;
default=0
    ;; opts("excludeRel","many")=1 => Relationship property of type 'many' does not constitute a dependency;
default=1
    ;; opts("buildErrorProject")="FooBar" => Create/recreate Studio project FooBar, consisting of items that
reported parse errors; default="" => no error project created
    ;;
    ;; OUTPUTS:
    ;; $$project = "" if success, else nnn\message
```

Note that by using this API you are agreeing that the output file it generates will only be used with Structure101g Studio, for which you must hold a Yuzinji flavor license.

4.4   Interpreting Results

Icons

The Yuzinji flavor uses the following icons in Structure101g Studio:

	Package
	Class
	Abstract class
	Datatype class
	Method of a class
	Empty method, with no lines of code. For metrics purposes it is treated as having one line, so that it adds to the complexity of the class.
	Property of a class
	Query of a class
	Index of a class
	Foreign key
	Routine
	Line label within a routine
!	Error during parsing

Dependency Types

Yuzinji detects and reports the following types of dependency:

is-of-type	Item is of datatype
extends	Class extends class
has-FK-to	Foreign key uses index
is-part-of	Property is part of foreign key
returns	Method returns datatype
has-param	Method has parameter of datatype
calls-label	Code calls line label as subroutine
jumps-to	Code jumps to line label
evaluates-function	Code calls line label as extrinsic function
invokes-method	Code invokes classmethod
queries	Code queries the data of another class through SQL
jobs	Code JOBs line label in routine
xcalls	Code invokes XCALL - DO &pkg.func(arg1,arg2)
(uses)	Other usage

Stub Items

For the completeness of the model, a method of a class may need to be displayed even if there was no class-specific code to analyze (i.e. the method is implemented by a superclass). For example, a %OpenId classmethod call to it was detected within other code. When this occurs the method is given a small red questionmark adornment. For example, in the screenshot below the %New and %OpenId methods are occurrences of this. The adornment propagates upwards onto the AddressPersistent class and its parent Data package, with a faded appearance unless all descendants possess the same adornment.
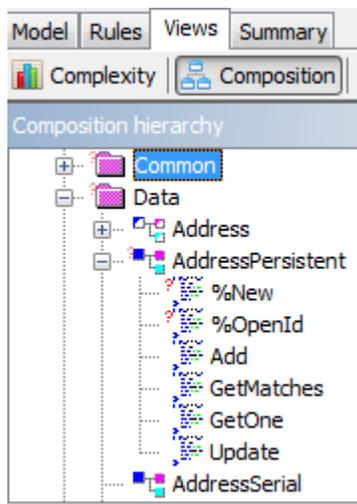
Parse Errors

If the Yuzinji parser encounters syntax it is unable to handle, information is added to the file sent to Structure101g Studio. This is presented in two ways within the UI. First, the class or routine containing the parse error is adorned with a white X in a red square. This adornment propagates upwards, with a faded appearance unless all descendants possess the same adornment. Second, an entry appears under a top-level [~Parse Errors] pseudo-package, giving details of the unparsed line. In the following screenshot a parse error in %SYS.Audit is illustrated.
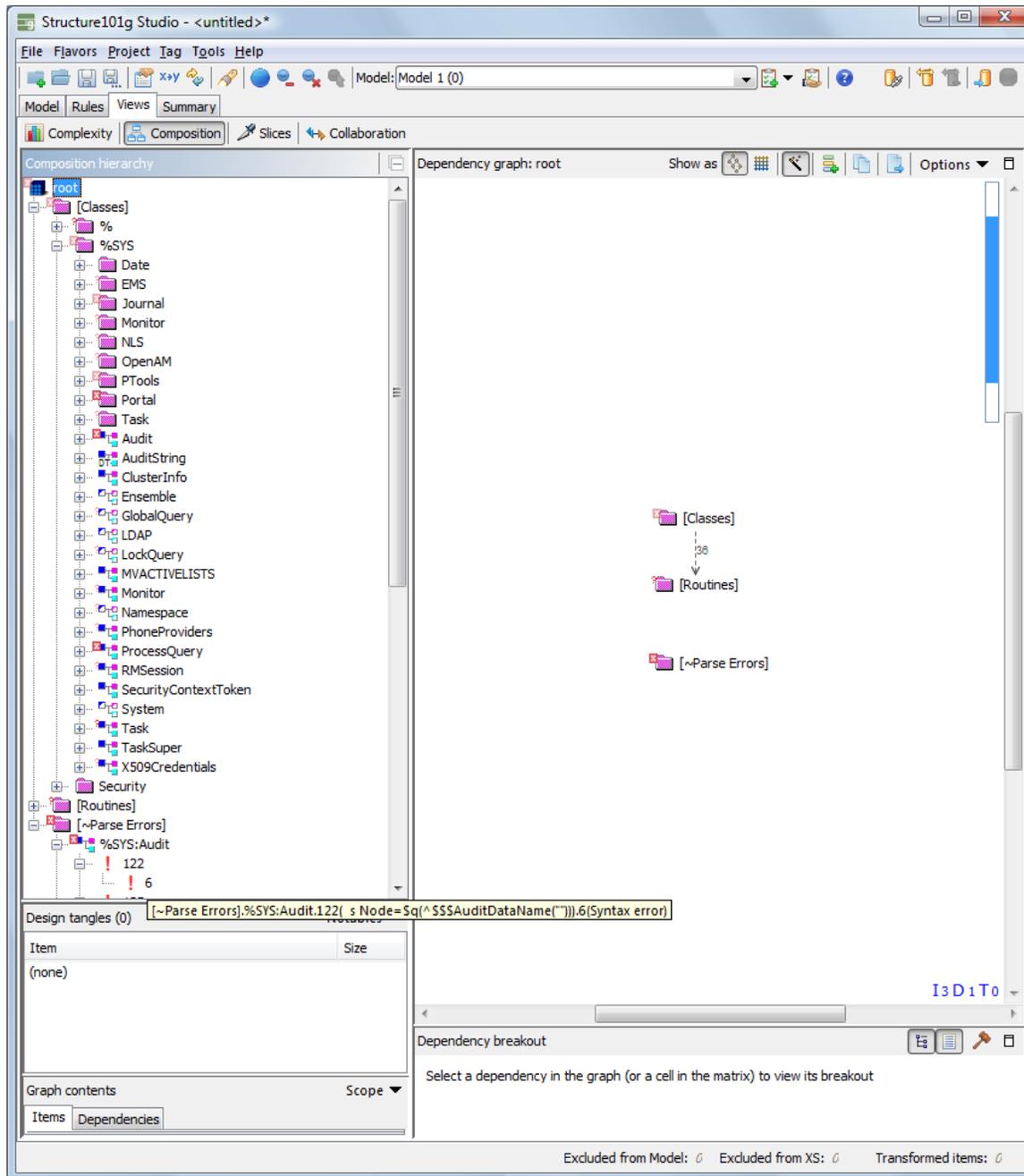
5 Updating Yuzinji

The Flavors menu of Structure101g Studio includes the option to check for updates. If an update for Yuzinji is found, select it and continue. This will update the flavor (plug-in) in Structure101g Studio. Then perform the following steps:

1. Close Structure101g Studio.

2. Start Structure101g Studio again.

3. From its menu select Help\About.

4. Select the Licensed Flavors tab.

5. The **Install dir** value tells you where to find the files used in subsequent steps. If you want to you can select the path text, copy it to the system clipboard (e.g. using Ctrl+C), then paste it into your directory browsing tool (e.g. Windows Explorer).

Now you need to update the server-side component of Yuzinji on any InterSystems server where it was previously installed.

6. Use System Management Portal to discover the path to your existing YUZINJI database (e.g. using System Operation > Databases). It will normally be **devuser\studio\templates\yuzinji** under your installation directory.

7. Shut down the InterSystems server instance. We recommend this shutdown to avoid potential issues with processes continuing to run previous Yuzinji code even after the update. If such a shutdown is impractical, dismount the YUZINJI database at this step.

8. Delete the **yuzinji** directory identified in step 6 above, together with all contents and subdirectories.

9. In the **serverkits** subdirectory of the Yuzinji install dir (as determined in step 5 above), choose the zipfile matching your server version. For instance:

    InterSystems2010.2plus-yuzinji1.2.0.zip

is suitable for InterSystems Caché and Caché-based platforms (e.g. Ensemble) version 2010.2 or greater, and for InterSystems IRIS.

10. Unzip this file, which gives you a **yuzinji** subdirectory.

11. Move the subdirectory created by the unzip into the **devuser\studio\templates** subdirectory of your InterSystems installation, reinstating the **yuzinji** directory you deleted in step 8 above.

12. Start the instance if it was shut down in step 7 above. If you only dismounted the YUZINJI database at that step, remount it now.

13. Start a Studio or Atelier session. Connect as a user with sufficient privileges to amend the configuration. These privileges are only required the first time you run the Yuzinji add-in after installing the update.

14. Connect to a namespace that you previously used Yuzinji in.

15. On the "Tools" menu choose "Add-Ins\Add-Ins..." then click the **Refresh** button on the dialog that appears. If you are using Atelier and do not have a **Refresh** button, run the following command in a terminal session in %SYS instead:

    ```
    d ##class(%CSP.StudioTemplateMgr).UpdateTemplates()
    ```

16. When the refresh has completed, choose the Yuzinji add-in and click **OK**.

17. The Yuzinji update is now complete on this server instance. You can proceed with a Yuzinji analysis immediately, or you can close the add-in and use it later.

6   Uninstalling the Server-side Component

To remove Yuzinji's server-side component from an InterSystems server instance, perform the following steps.

In System Management Portal (System>Configuration>Namespaces) determine which namespaces have been Yuzinji-enabled. There will be a partner YUZ-namespace for each enabled namespace.

For example:

The SAMPLES namespace has been Yuzinji-enabled, as indicated by the YUZ-SAMPLES namespace.

If the %SYS namespace was enabled its partner namespace would be named YUZ-_SYS because the % character is only permitted at the start of a namespace name.

For each Yuzinji-enabled namespace, delete the following package mapping:

Don't forget to click Portal's "Save Changes" button to commit the deletion.

Now go to System>Configuration>Local Databases and delete each YUZ-xyz database:

Check the boxes that will remove the associated YUZ-xyz namespace plus the actual YUZ-xyz database file:
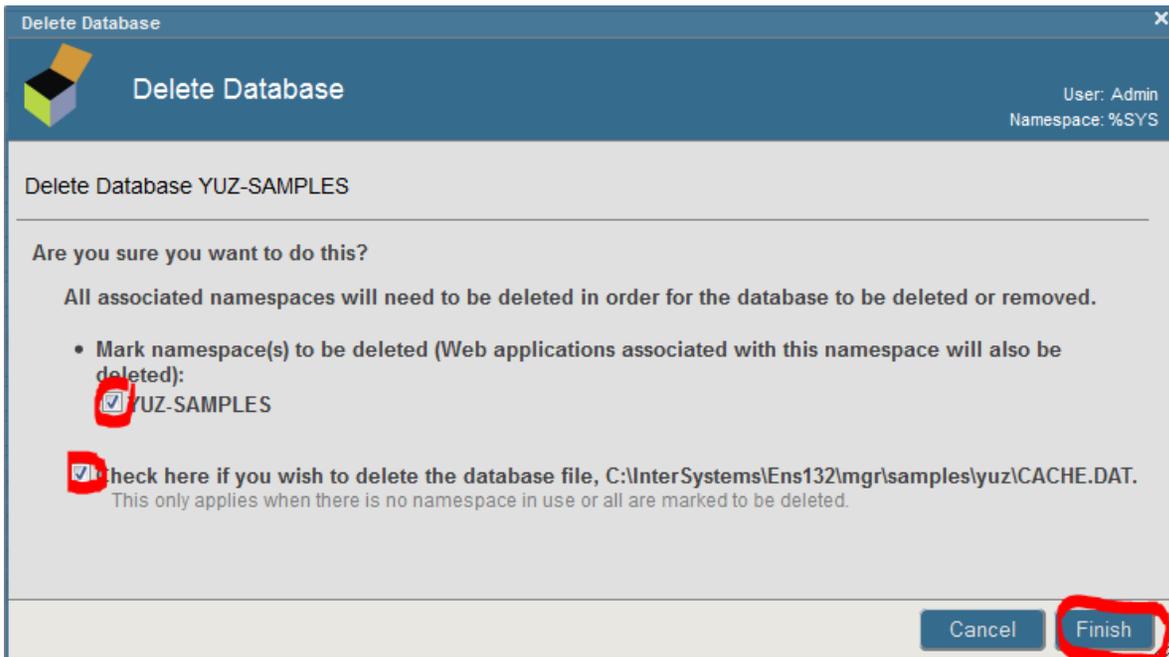
For maximum tidiness, delete the the remnants of the **yuz** directory that contained the database you just removed. The full path was shown on the previous dialog.

When no YUZ-xyz databases remain in the list, delete the YUZINJI database:

Check the boxes to permit deletion of the associated namespace, web application, and database file:
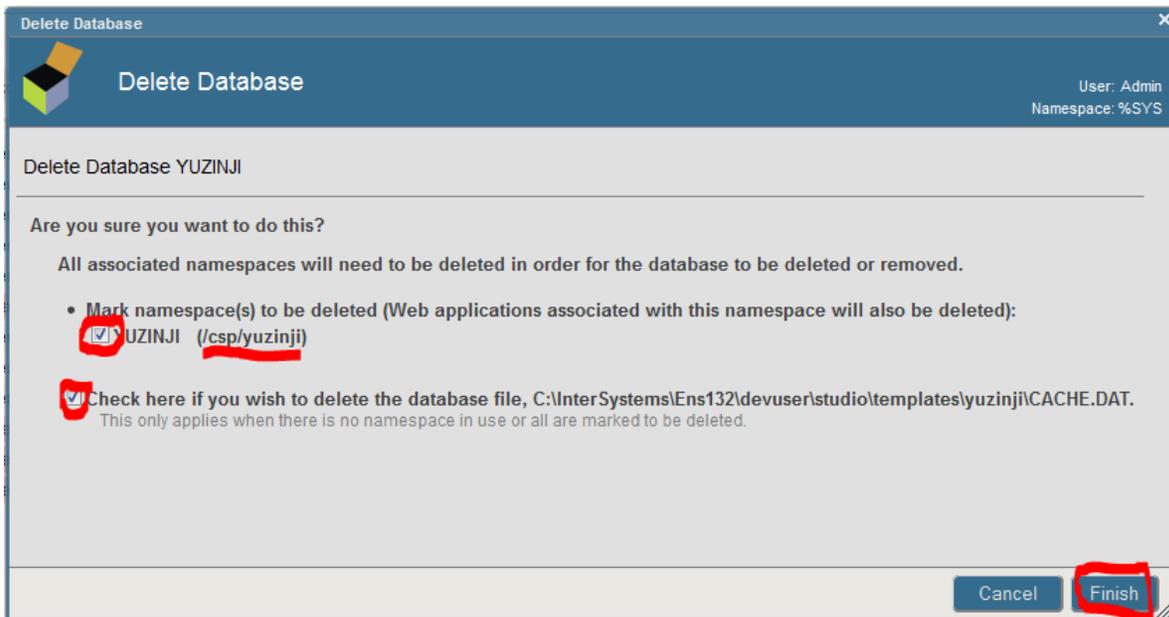
Delete the **yuzinji** subdirectory that you originally placed in the devuser\studio\templates\ subdirectory of your InterSystems server installation. The full path was shown to you on the previous dialog when you confirmed deletion of the database file within it.

Using Portal or Studio, delete the %Z.usertemplate.yuzinji package from the %SYS namespace.