

Developing on the Sandbox

A team's applications in ScienceCloud can be extended and customized using user-provided protocols and components. The Sandbox provides both an *authoring environment* for creating or editing these protocols, and a *test environment* to publish and test those protocol within their applications. These application protocols and components are saved on the protocols tab, in folders of the format:

Protocols/ScienceCloud/<team>/<application>

Since the Sandbox server is designated for authoring protocols for target ScienceCloud applications, you can only save protocols in a specific area of the Protocols tab designed for this purpose. Ensure you place your protocols for testing into the appropriate folder tree for the target application. You do not have permission to save a protocol elsewhere in the Protocols tab, only within your team folder under the *ScienceCloud* folder.

Each application has its own folder (and possibly subfolders) for their protocols. For example, if you write a protocol for use with the Mobile app, place the protocol in the *Mobile* folder hierarchy, which is rooted at *Protocols/<team>/Mobile*.

Within each of the application folders, certain organizational conventions might apply. For example, the Mobile app only supports two levels of subfolders below the root folder, so it can include a number of "category" folders. All protocols should be saved into one of these categories.

Application Folder	Subfolder 1	Note
Project Data	Tasks	Protocols will appear in a menu called "Tasks" in the Project Data application.
Project Data	Registration	Protocols will be used by Accelrys to customize the registration procedure in the Project Data application (for example, through custom business rules, salt stripping, or structure validation).
Web Tasks	<Defined by developer>	Protocols will appear in the Web Task application either at the top level or in a folder that you create.
Pipette Analysis	<Defined by developer>	Protocols will appear in the Pipette Analysis application. You can organize protocols in folders on the Sandbox, but the application will always present the protocols in the application as a flat list.
Mobile	<Defined by developer>	Protocols will appear in the Mobile App. Only one level of hierarchy is supported in the Mobile App.
Assay	Readers, Visualization Templates, etc	Protocols will appear in the Assay application.
Notebook	Tasks	Protocols will appear in the Notebook application.
Components	<Defined by developer>	Shared components that all application protocols can use.

(The “Components” folder is a special case. It is not an application; instead, it is an area where shared components can be saved and then used by application protocols. However, it will be treated like an application for the purposes of publication for testing.)

Initial testing of any protocol is achieved by running the protocol in the Pipeline Pilot client that is connected to the Sandbox server. We recommend performing most protocol testing this way, trying out different input options. Diagnosing and debugging problems in the Sandbox environment is generally easier than from any other client application.

You can create, save, and test application protocols using your Pipeline Pilot client, but to test these protocols within their application, you will need to *publish* them for Sandbox testing. (Later, when you are ready for production, you will publish them to ScienceCloud; that will be described later.)

Most, but not all, applications need publication for application testing. The applications that need publication prior to testing are: Assay, Components, Pipette Analysis, Notebook, Project Data, and Web Tasks.

Publication for Sandbox Testing

When your team’s protocols or components are ready for application testing on the sandbox, you will need to run either the protocol *Publish Team Protocols for Sandbox Testing* (to publish *all* the protocols of a team for testing) or protocol *Publish Single Protocol for Sandbox Testing* (to publish a single protocol for testing). Publication makes them available in the Sandbox’s applications.

Once published, your testing procedure will depend on the target application.

For example, for Project Data protocols, you would test by going to <https://sandbox.sciencecloud.com> in a browser, then click on the Application "Project Data".

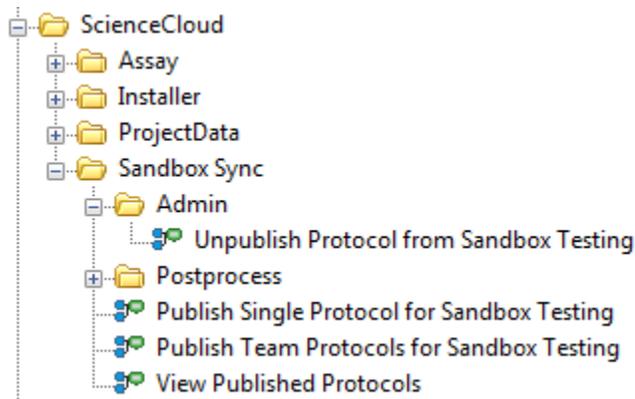
Another example would be testing Web Tasks. Go to <https://sandbox.sciencecloud.com> in a browser, then click on the Application "Web Tasks". You should be able to see and run your published web task.

Your application help documentation should describe the types of customizations possible, and the correct folder locations for different customizations.

The following protocols are available to assist you with the publication process:

- *Publish Team Protocols for Sandbox Testing*: publish any new or edited protocols for a particular team and a set of applications for Sandbox testing.
- *Publish Single Protocol for Sandbox Testing*: publish a single protocol or component for Sandbox testing.
- *View Published Protocols*: view all published protocols.
- *Unpublish Protocol from Sandbox Testing*: (Team Admin only): unpublish a published protocol or component so it is no longer available within its application for Sandbox testing.

You’ll find them in the “Sandbox Sync” folder in Protocols/Utilities/ScienceCloud:

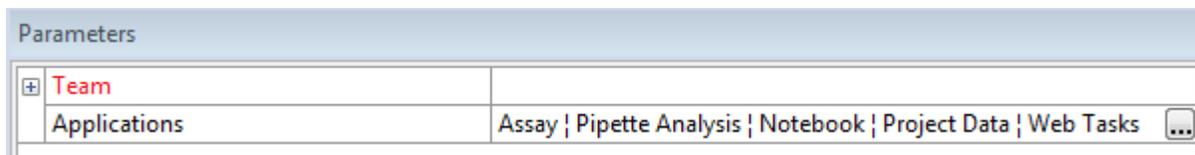


We will discuss each of these protocols in turn.

Protocol *Publish Team Protocols for Sandbox Testing*

This protocol allows a user on the Sandbox to deploy new or changed application protocols to the Sandbox for testing. This protocol is in: Protocols/Utilities/ScienceCloud/Sandbox Sync/.

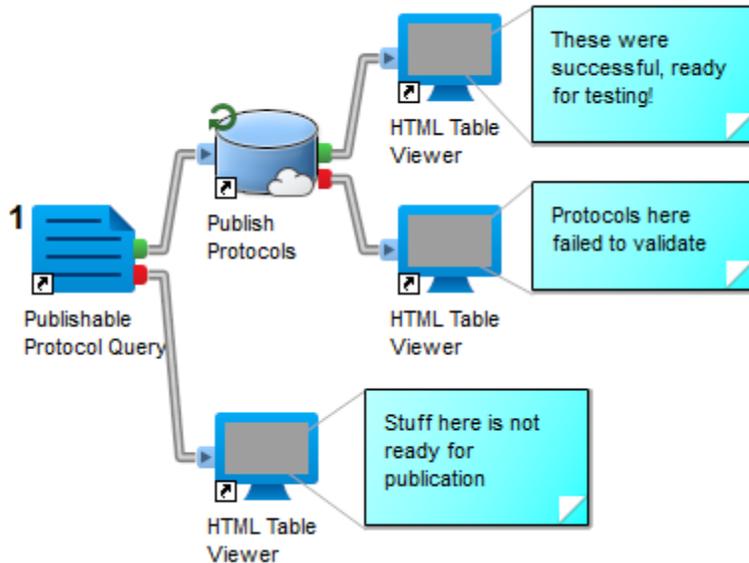
The interface of the protocol looks like this:



The parameters are:

- **Team:** the team whose subfolders contain the protocols and components that need publication. (Grouped beneath “Team” are the optional parameters “SC Username” and “SC Password”. You can provide these if you wish to execute as a different ScienceCloud user.)
- **Applications:** the application subfolders to check for new protocols or components. By default, all are selected; if you are working on a single application, you may wish to uncheck unnecessary application names.

The protocol looks like:



When you run this protocol, application protocols and components ready for publication are returned out the Pass port of the *Publishable Protocol Query* component. The *Publish Protocols* component will then validate them (that is, check they are suitable for publication), then publish them if they are.

Application protocols and components that need publication but cannot for some reason are returned out the Fail port, with annotations suggesting what actions your Team Admin can take to make things right. For example, you may have published a protocol in the past, then moved or deleted the protocol from the Sandbox. In this case, you'll need the assistance of the Team Admin to unpublish the previously published protocol before you proceed.

Protocols returned from the Pass port of *Publishable Protocol Query* are ready to publish. You hand them to the *Publish Protocols* component to do this. Upon successful publication, the record is returned out the Pass port.

If a protocol record is returned out the Fail port, then it either "Failed to Validate" and hence was not published; or it was published, but "Needs Further Action" - the user needs to do additional actions before testing.

If your protocol fails to validate, then you will need to edit it to fix the problem before publication can be accepted.

If your protocol needs further action, following any associated instructions.

Here is the output from attempting to publish three protocols: the first is successful, the second published, but needs further action, and the third failed to validate:

Path	Status	Message	RuleName	RuleMessage
Protocols/ScienceCloud /SCREGRESSCUST /Pipette Analysis/Outputs/HTML report	Ready for Testing			
Protocols/ScienceCloud /SCREGRESSCUST /Assay/Readers /TopCount Reader - TB	Needs further action	Assay needs additional configuration to work with published protocols and components. Please refer to "http://help.sciencecloud.com /customer/en/portal/articles /2309580-assay-user-guide" for more information.		
Protocols/ScienceCloud /SCREGRESSCUST /Assay/Readers/Reader with SQL Component	Failed to Validate		Design: Protocol contains disallowed component	Found the component SQL Select, which is forbidden in a multi-tenant context. (Display name is SQL Select.)

Protocol *Publish Single Protocol for Sandbox Testing*

This protocol is in: Protocols/Utilities/ScienceCloud/Sandbox Sync/.

This protocol allows a user on the Sandbox to deploy a new or changed protocol or component to the Sandbox for testing.

The interface of the protocol looks like this:

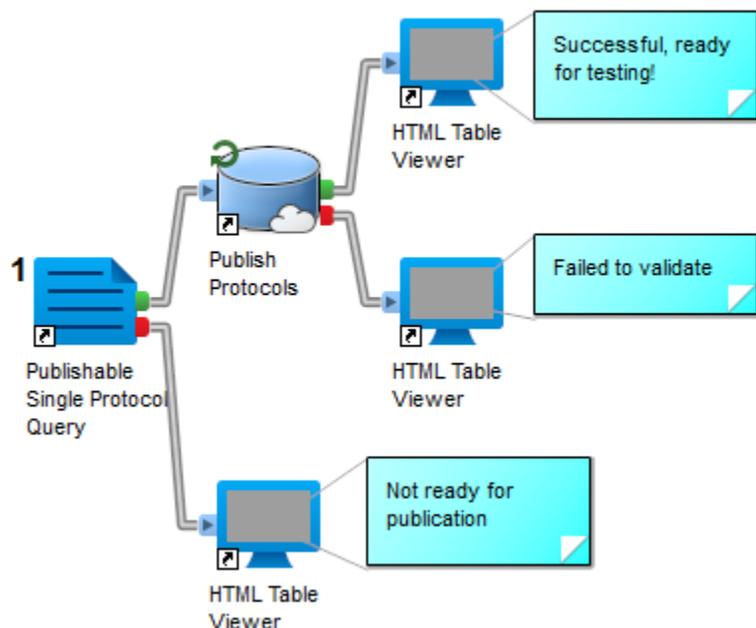
Parameters

+ Path	
--------	--

The parameters are:

- Path:** the path to the protocol or component, which must be in a subfolder of Protocols/ScienceCloud/<someteam>/<someapp>/ . (Grouped beneath "Team" are the optional parameters "SC Username" and "SC Password". You can provide these if you wish to execute as a different ScienceCloud user.)

The protocol looks like:



When you run this protocol, if the provided protocol or component is ready for publication, it will be returned out the Pass port of the *Publishable Single Protocol Query* component. The *Publish Protocols* component will then validate them, and if they pass, publish them.

Application protocols and components that need publication but cannot for some reason are returned out the Fail port, with annotations suggesting what actions your Team Admin can take to make things right. For example, you may have published a protocol in the past, then moved or deleted the protocol from the Sandbox. In this case, you'll need the assistance of the Team Admin to unpublish the previously published protocol from the Sandbox before you proceed.

Protocol View Published Protocols

This protocol is in: Protocols/Utilities/ScienceCloud/Sandbox Sync/.

This protocol allows a user to view all published protocols and components on the Sandbox.

The interface of the protocol looks like this:

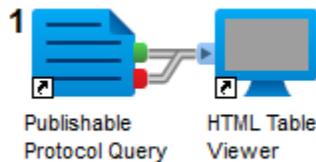
Parameters	
Team	
Applications	Assay Pipette Analysis Notebook Project Data Web Tasks ...

The parameters are:

- **Team:** the team whose subfolders contain the protocols and components that need publication. (Grouped beneath "Team" are the optional parameters "SC Username" and "SC Password". You can provide these if you wish to execute as a different ScienceCloud user.)

- **Applications:** the application subfolders to check for new protocols or components. By default, all are selected; if you are working on a single application, you may wish to uncheck unnecessary application names.

The protocol looks like:



When you run this protocol, the *Publishable Protocol Query* component will list the status of all protocols for the given Team and the selected Applications.

The following is an example of the output for the team SCREGRESSCUST:

Name	Path	RemoteSyncRegDate	RegDate	Status	RemotePath	ErrorText
TopCount Reader - TB	Protocols/ScienceCloud /SCREGRESSCUST /Assay/Readers/TopCount Reader - TB	Wed Jun 08 08:05:03 2016	Wed Jun 08 08:05:03 2016	Already Published		
Search with Pipette Sketcher	Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/Search with Pipette Sketcher	Wed Jun 08 08:05:05 2016	Wed Jun 08 08:05:12 2016	Different Registration Dates		
				Original Version Missing	Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/PUBLISH Search with Pipette Sketcher	The remote path points to a published protocol that no longer exists on the sandbox. Please send mail to your Team Admin to unpublish this for you.
Search with Pipette Sketcher	Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/Subfolder/Search with Pipette Sketcher			Location Changed Since Last Publication	Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/PUBLISH Search with Pipette Sketcher	The protocol at "Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/Subfolder /PUBLISH Search with Pipette Sketcher" was previously published, but at location "Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/PUBLISH Search with Pipette Sketcher". You will need to ask your Team Admin to unpublish the old published protocol.
Search with Pipette Sketcher	Protocols/ScienceCloud /SCREGRESSCUST/Web Tasks/Subfolder/Search with Pipette Sketcher		Wed Jun 08 08:05:14 2016	Not Yet Published		

The list will contain all protocols for the given team, along with their publication status. Common status codes are:

- **Not Yet Published:** this is a new protocol or component that has not yet been published. You'll need to publish it if you wish to do application testing.

- **Different Registration Dates:** the local Sandbox version of the protocol or component is newer than the current published version. You need to publish again if you wish to use this newer version in your application testing. If you do not publish, the application will continue to use the previous published version.
- **Already Published:** this is a protocol or component that has already been published, and the published version is fully up-to-date. (By default, the *Publishable Protocol Query* component has this Return option turned off – you’ll need to turn this on to see these entries.)
- **Location Changed Since Last Publication:** you have moved this protocol or component since it was last published. You’ll need to contact your Team Admin to remove it from the Sandbox before you can continue to publish it in its new location.
- **Original Version Missing:** there is a published protocol or component, but no corresponding protocol or component on the authoring server was found. This can happen if you delete the protocol or component in the Pipeline Pilot client after previously saving and publishing. You’ll need to contact your Team Admin to unpublish it.

Only two of these status codes mean your protocol or component is ready for publication. These are: “Not Yet Published” (for previously-unseen protocols) and “Different Registration Dates” (for previously published protocols that have been updated). These are returned out the Pass port. All others are not ready for publication (or are already published), and are returned out the Fail port.

Protocol *Unpublish Protocol from Sandbox Testing*

This protocol is in: Protocols/Utilities/ScienceCloud/Sandbox Sync/Admin/.

Sometimes, you'll need to unpublish published protocols or components from the Sandbox. For security, only the Team Admin is allowed to perform this operation.

This Team Admin protocol allows the team admin on the Sandbox to unpublish a particular protocol.

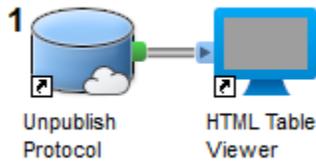
The interface of the protocol looks like this:



The parameters are:

- **Path:** the path to the protocol or component as published, which must be in a subfolder of Protocols/ScienceCloud/<someteam>/<someapp>/. Typically, you’ll get this from the property “RemotePath” that the user will provide from *View Published Protocols* Fail port output. (Grouped beneath “Team” are the optional parameters “SC Username” and “SC Password”. You can provide these if you wish to execute as a different ScienceCloud user.)

The protocol looks like:



When you run it, the protocol or component specified by Path will be unpublished. Note that this does NOT affect any component or protocol on the Sandbox. You're free to publish it again!

Typically, you get the Path parameter from the value of "RemotePath" when the user reports the error:

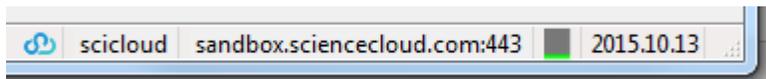
Name	Path	Status	RemotePath	ErrorText
Registration Protocol	Protocols/ScienceCloud /SCREGRESSCUST /Project Data/Registration /Registration Protocol	Location Changed Since Last Prepublication	Protocols/ScienceCloud /SCREGRESSCUST /Project Data/Test /Registration Protocol	The protocol at "Protocols/ScienceCloud /SCREGRESSCUST /Project Data/Registration /Registration Protocol" was previously prepublished, but at location "Protocols/ScienceCloud /SCREGRESSCUST /Project Data/Test /Registration Protocol". You will need to ask your Team Admin to delete the old prepublished protocol.

By cutting and pasting the value of RemotePath into the Path parameter of the protocol, the Team Admin can unpublish it, removing it from its application for Sandbox testing.

Example: Authoring and Testing a Web Task

Let's go through the process of authoring and testing a Web Task on the Sandbox.

We start by opening a Pipeline Pilot client and connecting to `sandbox.sciencecloud.com`. Click on your server name in the lower right corner and enter that as the server name. Once you connect, it should look like:

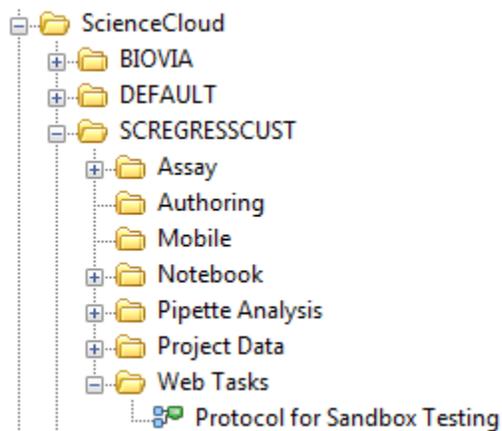


Now, open a New Protocol and add an *MAO Reader* and an *HTML Table Viewer*:



Run the protocol to make sure it works. Beautiful!

Now, save the protocol in folder *Protocols/ScienceCloud/<yourteamname>/Web Tasks* as *Protocol for Sandbox Testing*:

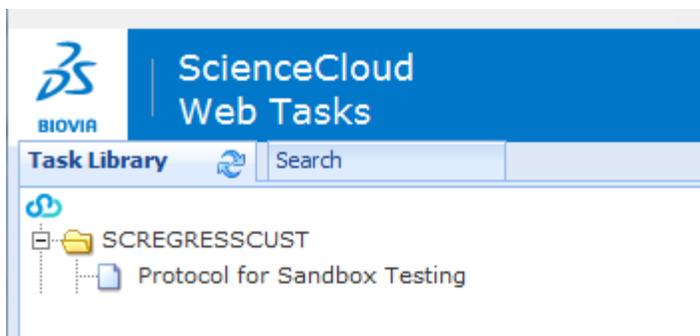


We're ready to publish now. Open the protocol *Publish Team Protocol for Sandbox Testing*. In the protocol parameter "Path" enter the value *Protocols/ScienceCloud/<yourteamname>/Web Tasks/Protocol for Sandbox Testing*. Run the protocol.

A browser window will open with something similar to the following:

Published and Ready for Testing	
Path	Status
Protocols/ScienceCloud/SCREGRESSCUST/Web Tasks/Protocol for Sandbox Testing	Ready for Testing

You're ready to test in the application now. In a browser, go to <https://sandbox.sciencecloud.com> and login. Under "Apps", select "Web Tasks". You should see your new protocol, ready for testing.



If you make a change to the original protocol and wish to test that update in the Web Tasks application, you'll need to rerun *Publish Team Protocol for Sandbox Testing* after making your edits.

Finally, while you are allowed to update published protocols, only the Team Admin can delete them. It's a good idea to know who your team admin is in advance, so you can contact them if an application protocol needs to be moved, renamed, or removed.