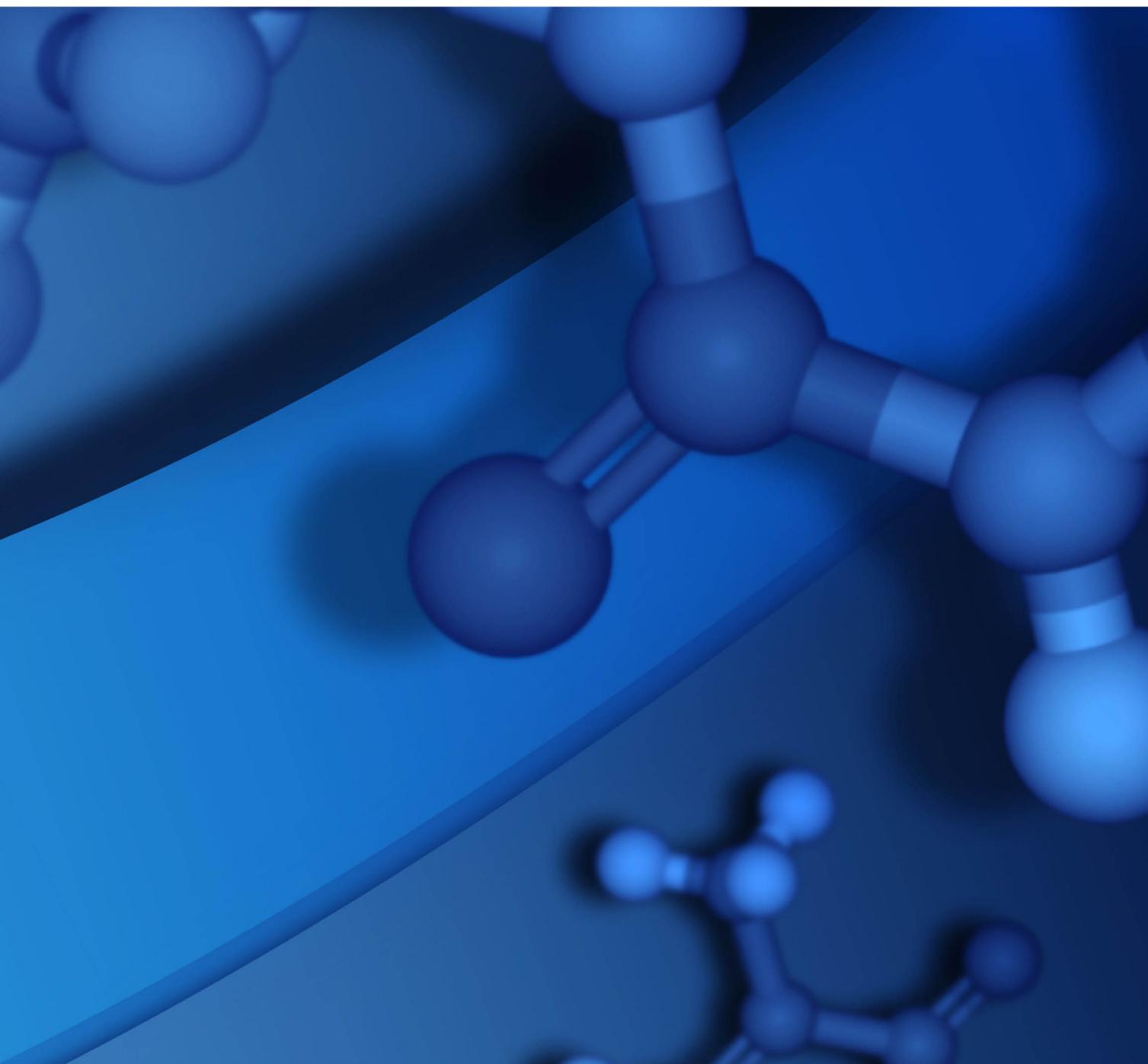


PIPELINE PILOT INTEGRATION GUIDE

BIOVIA WORKBOOK 2021



Copyright Notice

©2020 Dassault Systèmes. All rights reserved. 3DEXPERIENCE, the Compass icon and the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3DVIA, 3DSWYM, BIOVIA, NETVIBES, IFWE and 3DEXCITE, are commercial trademarks or registered trademarks of Dassault Systèmes, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the U.S. and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

Acknowledgments and References

To print photographs or files of computational results (figures and/or data) obtained by using Dassault Systèmes software, acknowledge the source in an appropriate format. For example:

"Computational results were obtained by using Dassault Systèmes BIOVIA software programs. BIOVIA Workbook was used to perform the calculations and to generate the graphical results."

Dassault Systèmes may grant permission to republish or reprint its copyrighted materials. Requests should be submitted to Dassault Systèmes Customer Support, either by visiting <https://www.3ds.com/support/> and clicking **Call us** or **Submit a request**, or by writing to:

Dassault Systèmes Customer Support
10, Rue Marcel Dassault
78140 Vélizy-Villacoublay
FRANCE

Contents

BIOVIA Workbook and Pipeline Pilot Integration	1
In Vivo Experiment Example	1
Formulations DOE Experiment Example	6
Dose Response Experiment Example	7
Synthetic Chemistry Experiment Example	10
Interactive Overview	11
Dynamic Toolbar Item and RunProtocol Permissions	11
Integration Architecture	12
Build Templates that Integrate with Pipeline Pilot	12
Create an Experiment Template to Run a Pipeline Pilot Protocol	13
Prerequisites	14
Configuration steps	14
Test the Capability to Run the Protocol	15
Enable Changes to Runtime Values	16
Script Modifications to Customize the Integration	17
Processing Data for Export to Pipeline Pilot	17
Accessing data in an external system	18
Modifying Parameter Values	18
Post-processing Pipeline Pilot Data	18
.NET APIs and IronPython Script	19
Authoring Protocols Overview	20
License Requirements for Workbook Specific Pipeline Pilot protocols	20
Development Process for Pipeline Pilot Protocols	20
Pipeline Pilot Protocols Organization	20
Pipeline Pilot Protocols Guidelines	21
Notebook Readers and Writers	21
Table Section CSV Format	22
Configure the Protocol	23
Troubleshooting	23
Data import issues	23
Format inconsistencies	23
Protocol issues	24
Salt stripping problems	24

Limitations with Previous Versions	25
Salts.sd	25
Limitations for integration with Pipeline Pilot	25

BIOVIA Workbook and Pipeline Pilot Integration

Workbook section data can be processed using Pipeline Pilot protocols, and the data and reports from the protocol can be imported back into the Workbook sections for use in the experiment.

BIOVIA Workbook includes experiment template examples that demonstrate how to use Pipeline Pilot protocols for such purposes.

Pipeline Pilot includes a Notebook collection with sample protocols for integrating with BIOVIA Workbook. This collection includes the examples described in this document. The sample Pipeline Pilot protocols provide a starting point for the kinds of tasks that end users can accomplish in BIOVIA Workbook. Benefits of the Pipeline Pilot protocols include:

- Data aggregation.
- Immediate access to volumes of disparate research data locked in silos.
- Automation of the scientific analysis of that data.
- Rapidly exploring, visualizing, and reporting research results.

By using Pipeline Pilot protocols, BIOVIA Workbook users can manage and synthesize scientific information in a way that allows them to rapidly adjust workflows, optimize research cycles, and respond to changing market conditions.

The following BIOVIA Workbook experiment templates demonstrate integration with Pipeline Pilot:

- [In Vivo Experiment](#)
- [Formulations DOE Experiment](#)
- [Dose Response Experiment](#)
- [Synthetic Chemistry Experiment](#)

In Vivo Experiment Example

The *In Vivo Experiment* example uses the following Pipeline Pilot

- Pareto scoring

This is used to assign animals to groups by minimizing Mean and StdDev variance among groups. The protocol starts by creating a number of initial random assignments, after which at each iteration the previous assignments are mutated by swapping the assignments for two members within the set. This results in double the number of assignments. All these are scored using Pareto scoring, and the best half is saved. Once all the iterations are complete, the result is a number of different assignments equal to the number of iterations times the number of initial assignments. From these, one assignment is selected using Pareto scoring for the whole set.

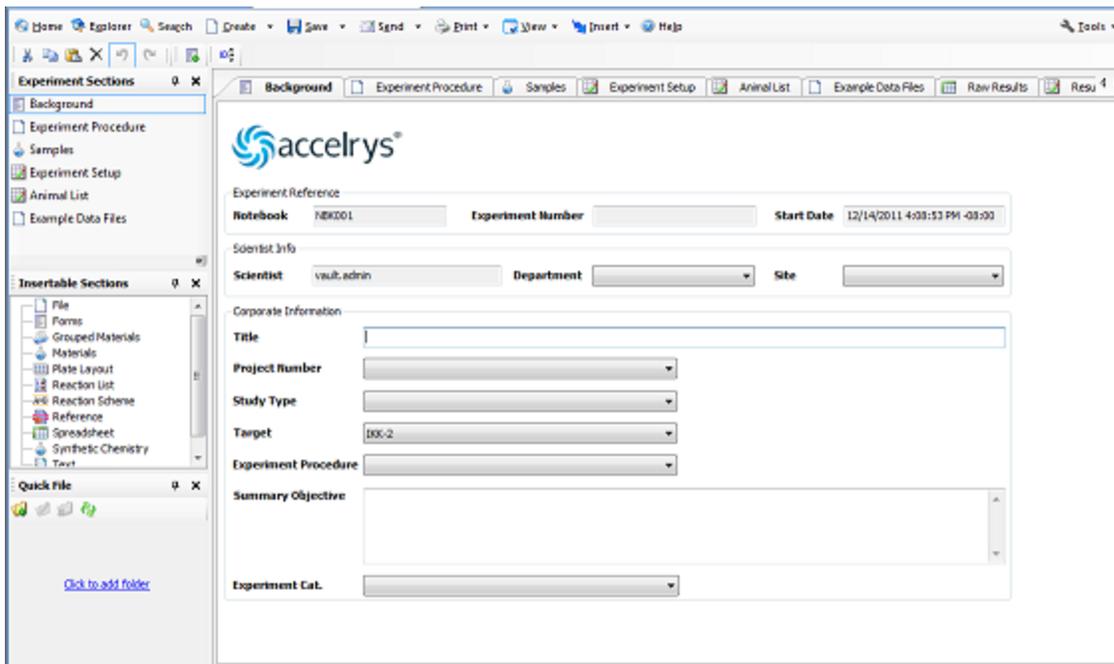
- Process Results

Uses data from a file on disk. The raw data and the calculated results are presented in their respective sections.

The *In Vivo Experiment* example measures the degree to which a specific transcription protein is inhibited in animals *In Vivo*. The target IKK-2, a protein serine kinase containing leucine zipper and helix-loop-helix protein interaction motifs, is a component of the large multiprotein complex I κ B kinase (IKK) signalsome, which is involved in the activation of the transcription factor nuclear factor kappa B (NF- κ B). The other targets available are Ghrelin, hERG, CCR5, and CCR7.

The *In Vivo Experiment* contains the following experiment sections:

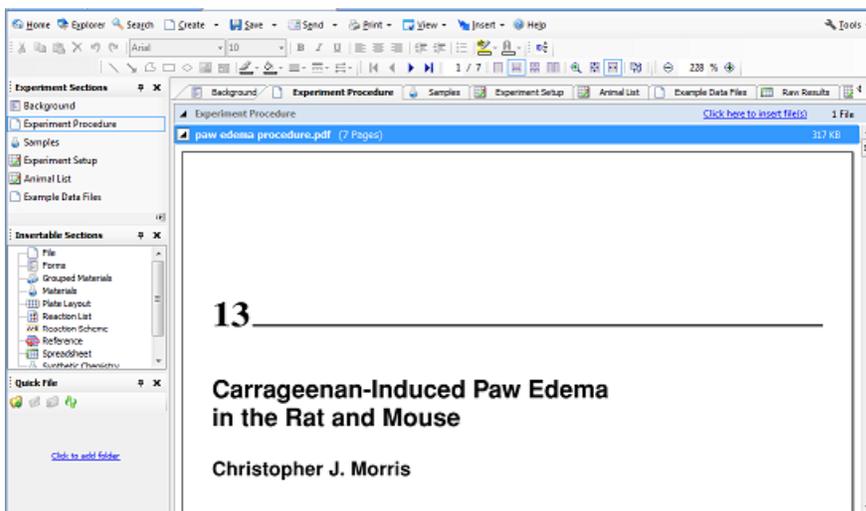
- The Background section of the experiment is a Forms section containing widgets that allow you to enter general information about the experiment.



The screenshot shows the BIOVIA software interface. The main window displays the 'Background' section of an experiment. The interface includes a menu bar (Home, Explorer, Search, Create, Save, Send, Print, View, Insert, Help) and a toolbar. On the left, there are panels for 'Experiment Sections' (Background, Experiment Procedure, Samples, Experiment Setup, Animal List, Example Data Files), 'Insertable Sections' (File, Forms, Grouped Materials, Materials, Plate Layout, Reaction List, Reaction Scheme, Reference, Spreadsheet, Synthetic Chemistry, Text), and 'Quick File' (Click to add folder). The main content area shows the 'Background' section with the 'accelrys' logo and the following form fields:

- Experiment Reference: Notebook (NBR001), Experiment Number, Start Date (12/14/2011 4:03:53 PM -08:00)
- Scientist Info: Scientist (vault.admin), Department, Site
- Corporate Information: Title, Project Number, Study Type, Target (BOS-2), Experiment Procedure, Summary Objective, Experiment Cat.

- The Experiment Procedure section of the *In Vivo Experiment* example is a File section containing the file *paw edema procedure.pdf*.



- You can add and delete files and graphical images in the Experiment Procedure section.

Name	Structure	Lot #	Sample ID	Density	MF	MW
1 CBAE-003529		1			C ₂₂ H ₂₈ N ₄ O ₅ S	479.6
2 CBAE-004093		2			C ₁₃ H ₂ Cl ₂ N ₂ O	279.1
3 CBAE-006228		1			C ₁₃ H ₁₁ Cl ₂ N ₂ O ₂	312.2
4 CBAE-008532		3			C ₁₉ H ₁₁ ClF ₂ N ₃ O ₂ S	453.8
5 CBAE-001861		1			C ₁₂ H ₂ F ₂ N ₄ O ₂ S	419.4

- The Samples section of the *In Vivo Experiment* example is a specific type of Table section that enables entering information about each sample, including its structure.

Name	Structure	Lot #	Sample ID	Density	MF	MW
1 CBAE-003529		1	CBAE-003529-1		C ₂₂ H ₂₈ N ₄ O ₅ S	479.6
2 CBAE-004093		2	CBAE-004093-2		C ₁₃ H ₂ Cl ₂ N ₂ O	279.1
3 CBAE-006228		1	CBAE-006228-1		C ₁₃ H ₁₁ Cl ₂ N ₂ O ₂	312.2
4 CBAE-008532		3	CBAE-008532-3		C ₁₉ H ₁₁ ClF ₂ N ₃ O ₂ S	453.8
5 CBAE-001861		1	CBAE-001861-1		C ₁₂ H ₂ F ₂ N ₄ O ₂ S	419.4

- The Experiment Setup section of the *In Vivo Experiment* example is a specific type of Table section that allows you to enter information about the setup of the experiment. The Group ID values are generated in the Animal list using a Pipeline Pilot protocol.

Group ID	Group Size	Treatment	Agent ID	Agent Vehicle	Agent Route	Sample ID	Drug Dose	Drug Vehicle	Drug Route	Group Comments
1 A	10	Test	Carageenan	Tween, MC	PO	CBAE-003529-1	50 mg/kg	Phosal	IV	
2 B	10	Test	Carageenan	Tween, MC	PO	CBAE-003529-1	25 mg/kg	Phosal	IV	
3 C	10	Test	Carageenan	Tween, MC	PO	CBAE-004093-2	25 mg/kg	Phosal	IV	
4 D	10	Test	Carageenan	Tween, MC	PO	CBAE-006225-1	25 mg/kg	Phosal	IV	
5 E	10	Test	Carageenan	Tween, MC	PO	CBAE-008532-3	25 mg/kg	Phosal	IV	
6 F	10	Test	Carageenan	Tween, MC	PO	CBAE-008532-3	10 mg/kg	Phosal	IV	
7 G	10	Test	Carageenan	Tween, MC	PO	CBAE-001861-1	25 mg/kg	Phosal	IV	
8 H	10	Test	Carageenan	Tween, MC	PO	CBAE-001861-1	10 mg/kg	Phosal	IV	
9 I	10	Vehicle	Carageenan	Tween, MC	PO			Phosal	IV	
10 J	10	Control	Carageenan	Tween, MC	PO	REFERENCE	10 mg/kg	Phosal	IV	

- The Animal List section of the *In Vivo Experiment* example is a specific type of Table section that allows you to enter information about the animals in the experiment.

Animal ID	Animal Weight	Animal Age	Animal Gender	Group ID
1 ACC-0548	226 g	25 wk	male	
2 ACC-0864	212 g	28 wk	male	
3 ACC-0979	228 g	18 wk	male	
4 ACC-0980	215 g	9 wk	male	
5 ACC-0073	240 g	33 wk	male	
6 ACC-0291	214 g	24 wk	male	
7 ACC-0231	227 g	22 wk	male	
8 ACC-0054	227 g	12 wk	male	
9 ACC-0966	218 g	12 wk	male	
10 ACC-0127	219 g	20 wk	male	
11 ACC-0905	222 g	6 wk	male	
12 ACC-0962	225 g	17 wk	male	
13 ACC-0191	242 g	12 wk	male	
14 ACC-0767	232 g	16 wk	male	
15 ACC-0078	226 g	34 wk	male	
16 ACC-0854	201 g	31 wk	male	
17 ACC-0430	236 g	24 wk	male	
18 ACC-0986	220 g	16 wk	male	
19 ACC-0669	227 g	36 wk	male	
20 ACC-0408	214 g	16 wk	male	
21 ACC-0390	224 g	32 wk	male	
22 ACC-0682	232 g	31 wk	male	
23 ACC-0808	233 g	17 wk	male	
24 ACC-0965	233 g	24 wk	male	
25 ACC-0514	245 g	18 wk	male	
26 ACC-0381	212 g	18 wk	male	
27 ACC-0219	251 g	10 wk	male	

When you click the toolbar item labeled **Assign Groups**, the Pipeline Pilot Client **Protocol Settings** dialog for the Assign Groups protocol is shown.

The **Protocol Settings** dialog enables specifying whether to assign the animals to groups by their weight or their age. The Assign Groups protocol uses *Pareto scoring* to assign animals to groups by minimizing Mean and StdDev variance among groups.

For more information about the algorithm used by the Assign Groups protocol, see the **Protocol Description** region in the Pipeline Pilot Client **Protocol Settings** interface.

When you click **OK**, you see the **Run Pipeline Protocol** dialog, which disappears after a successful execution is complete.

After running the Pipeline Pilot Assign Groups protocol, the Group ID values are displayed in the Animal List section. The Group ID values are also displayed in the Experiment Setup section.

	Animal ID	Animal Weight	Animal Age	Animal Gender	Group ID
▶ 1	ACC-0548	226 g	25 wk	male	J
2	ACC-0864	212 g	28 wk	male	I
3	ACC-0979	228 g	18 wk	male	I
4	ACC-0980	215 g	9 wk	male	G
5	ACC-0073	240 g	33 wk	male	A
6	ACC-0291	214 g	24 wk	male	E
7	ACC-0231	227 g	22 wk	male	C
8	ACC-0054	227 g	12 wk	male	D
9	ACC-0956	218 g	12 wk	male	G
10	ACC-0127	219 g	20 wk	male	H
11	ACC-0905	222 g	6 wk	male	D
12	ACC-0962	225 g	17 wk	male	F
13	ACC-0191	242 g	12 wk	male	D
14	ACC-0757	232 g	16 wk	male	B
15	ACC-0078	226 g	34 wk	male	D
16	ACC-0854	201 g	31 wk	male	I
17	ACC-0430	236 g	24 wk	male	A
18	ACC-0985	220 g	16 wk	male	A
19	ACC-0659	227 g	36 wk	male	C
20	ACC-0408	214 g	16 wk	male	D
21	ACC-0390	224 g	32 wk	male	B
22	ACC-0682	232 g	31 wk	male	D
23	ACC-0809	233 g	17 wk	male	I
24	ACC-0955	233 g	24 wk	male	I
▶ 25	ACC-0514	245 g	18 wk	male	I

You can group the values that are in any column of the table such as the Group ID values. To group the values:

1. Right-click a column heading, and then select Group By Box.
2. Drag the column heading to the gray area above the column, such as the Group ID column heading.
The rows are grouped by Group ID.

The Example Data Files section of the *In Vivo Experiment* example is a File section with the ability to display the raw experimental data, such as data collected from an instrument.

The Raw Results section and the Results Summary section use the saved file on disk as the basis for the calculated results whether or not you display the contents of your data file in the Example Data Files section. Save the file section, before you navigate to any other sections that use the data in the Pipeline Pilot protocol.

When you click the toolbar item labeled Process Results, the **Pipeline Pilot Protocol Settings** dialog for the Process Results protocol is shown. The Process Results dialog enables choosing the file that contains the raw data collected when this experiment was run. The Process Results protocol processes the data file, that you can view in the Raw Results section, and calculates the results summary in the Results Summary section. In this example, the data file is the file RawResults.csv.

When you click **OK**, you see the **Run Pipeline Protocol** dialog that disappears after a successful execution is complete. Upon completion of the Pipeline Pilot Process Results protocol, the Raw Results section displays the % Inhibition values and their Delta values for the animals tested. The Raw Results section of the *In Vivo Experiment* example is a Spreadsheet section.

In addition, the Results Summary section displays the Mean % Inhibition and Mean Body Weight values with their respective statistical deviations for the groups of animals tested.

If you move your mouse over the Group ID C cell that contains the warning icon, you can see that the value is out of range for the standard error of the mean.

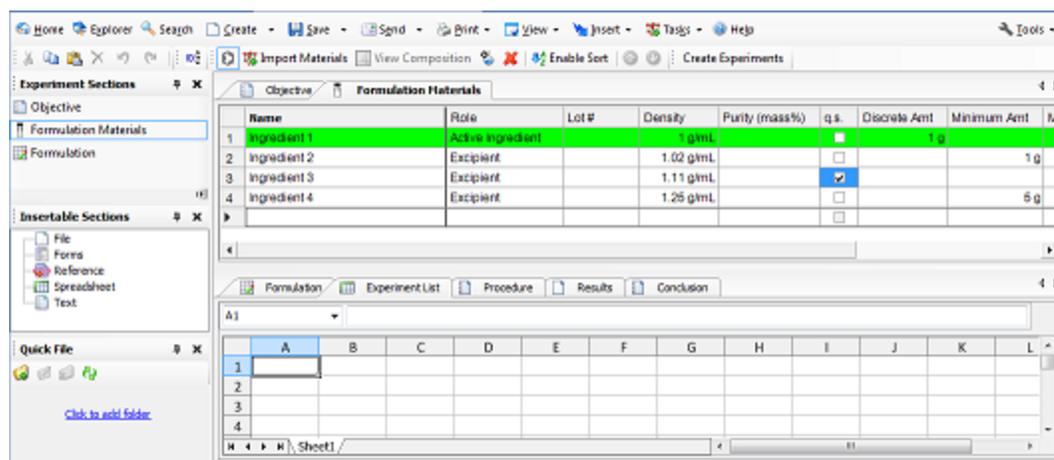
If you click the warning icon, you see a red background for that cell.

Formulations DOE Experiment Example

The *Design of Experiment (DOE)* example uses the Pipeline Pilot protocol Create Formulations. The example takes as input a list of materials and amounts specified in one of the following ways:

Value	Description
q.s.	Specifies the quantity sufficient to bring the formulation to the desired batch size. q.s is the balance material designation check box. Check the box to designate the ingredient as q.s.
Discrete Amt.	Specifies the fixed amount of ingredient for all formulations designed. Enter a specific amount with units as appropriate.
Minimum Amt.	Specifies the minimum amount of ingredient to use in the designed formulations. Enter a minimum amount for the range.
Maximum Amt.	Specifies the maximum amount of ingredient to use in the designed formulations. Enter a maximum amount for the range.
# of Variations.	Specifies the number of incremental steps for an ingredient amount with values of minimum (Minimum Amt.) and maximum (Maximum Amt.). If you enter a value of 5, then there are five different concentrations for the range that you specified in Minimum Amt. and Maximum Amt.

The protocol creates all the possible combinations as new formulations. The output is a spreadsheet containing the list of formulations and a hierarchical .csv file that can be imported into a Formulation section.



When you select **Create Experiments**, the Pipeline Pilot Client **Protocol Settings** dialog for the Create Formulations protocol is shown. The Protocol Settings dialog enables entering the desired batch size, `FormulationBatchSize`, and unit, `FormulationBatchSizeUnit`, for the experiments to generate. When you click **OK**, you see the **Run Pipeline Pilot Protocol** dialog, which disappears after a successful execution is complete. Upon completion of the Pipeline Pilot protocol Complete Formulations, the Formulation section enumerates the formulation experiments.

The Formulation section is populated with the amounts of each ingredient in each experiment. In this example, there are 20 experiments created that are ready for testing to determine the optimal formulation.

In addition, the experiment list is populated with the amounts of each ingredient in each experiment.

Experiment	Ingredient 1	Ingredient 2	Ingredient 3	Ingredient 4
Experiment 1	1 g	1 g	qs	5 g
Experiment 2	1 g	1 g	qs	8.3
Experiment 3	1 g	1 g	qs	11.0
Experiment 4	1 g	1 g	qs	15.0
Experiment 5	1 g	5.75 g	qs	5 g
Experiment 6	1 g	5.75 g	qs	8.3
Experiment 7	1 g	5.75 g	qs	11.0

Dose Response Experiment Example

The *Dose Response Experiment* uses the following Pipeline Pilot protocols:

- Plate Mapping

Takes as input a text file that contains samples that should be plated, as well as a number of parameters that control how the samples, dilutions, and replicates should be mapped on the desired-size plate. The output is a .pdf file with the sample and concentration plate layouts for each plate and an Excel spreadsheet with the well location information.

- Calculate EC50 Values

Converts sample concentrations to log scale, normalizes the response properly, gets the dose-response data for each unique compound, fits the dose-response data to a 4-parameter sigmoidal model, and displays the results in a report.

The *Dose Response Experiment* measures EC50 values using IKK-2 as the target in the experiment. The EC50 value is the half maximal effective concentration of a drug, antibody, or toxicant that induces a response halfway between the baseline and maximum after some specified exposure time.

IKK-2, a protein serine kinase containing leucine zipper and helix-loop-helix protein interaction motifs is a component of the large multiprotein complex I κ B kinase (IKK) signalsome that is involved in the activation of the transcription factor nuclear factor kappa B (NF- κ B). The other targets available are Ghrelin, hERG, CCR5, and CCR7.

The background section of the experiment is a forms section containing widgets that allow you to enter general information about the experiment. The assay procedure section of the *Dose Response Experiment* is a text section that contains the reagents, procedure, and equipment.

The screenshot shows the BIOVIA software interface. The top menu bar includes Home, Explorer, Search, Create, Save, Sign, Print, View, Insert, and Help. The toolbar contains icons for Home, Explorer, Search, Create, Save, Sign, Print, View, Insert, and Help. The sidebar on the left has two sections: 'Experiment Sections' and 'Insertable Sections'. 'Experiment Sections' includes Background, Assay Procedure, Samples, Plate Details, Protocols, Example Data Files, and Plate Layout. 'Insertable Sections' includes File, Forms, Grouped Materials, Materials, Plate Layout, Reaction List, Reaction Scheme, Reference, Spreadsheet, Synthetic Chemistry, and Text. The main content area displays the 'Background' form for an 'Experiment Reference'. The form includes fields for 'Notebook' (N890001), 'Experiment Number', and 'Start Date' (12/14/2011 4:08:31 PM -08:00). Below this is the 'Scientist Info' section with fields for 'Scientist' (vault.admin), 'Department', and 'Site'. The 'Corporate Information' section includes 'Title', 'Project Number', 'Study Type', 'Target' (39K-2), and 'Assay Procedure'. There is also a 'Summary Objective' text area and an 'Experiment Cat.' dropdown menu.

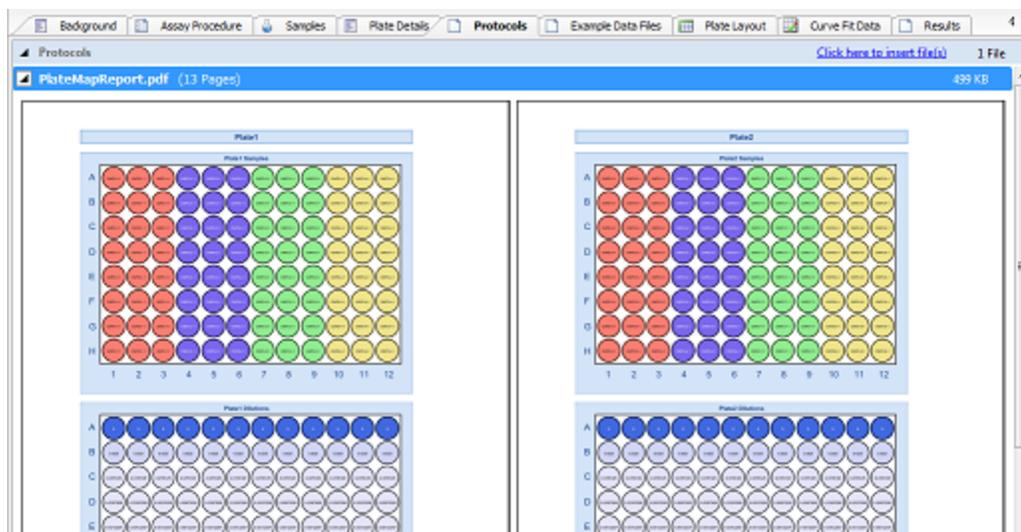
The samples section of the *Dose Response Experiment* is a specific type of table section that enables entering information about each sample, including information about the sample's structure.

After entering the sample information, generate Sample ID values. Click the toolbar item labeled ID. The Sample IDs are generated in the Sample ID column using the name of the sample with its lot number appended.

The Pplate details section of the *Dose Response Experiment* is a forms section that enables specifying the number of plates and their dimensions such as 8 rows by 12 columns, the number of dilutions of each sample and the dilution factor for the set of samples, the starting concentration of the set of samples, and the direction that the wells are assayed either down columns or across rows.

When you click the toolbar item Calculate Plate Layout, the Run Pipeline Pilot Protocol dialog for the Plate Mapping protocol is shown. The Plate Mapping protocol maps the plate and reports the results in the file PlateMapReport.pdf (in the Protocols section). Upon completion of the protocol execution, an output file (PlateMapReport.pdf) is created in the protocols section. In addition, this information populates the Spreadsheet in the Plate Layout section.

The Protocols section is a file section that contains the sample concentrations and their distributions on each plate.



The example data files section of the *Dose Response Experiment* is a file section with the ability to display the raw experimental data such as data collected from an instrument.

The plate layout section uses the saved file on disk such as the example `MicroBeta.txt` as the basis for the calculated results whether or not you display the contents of your data file in the example data files section. Save the file to disk before you navigate to any other sections that use the data in the Pipeline Pilot protocol.

The Plate Layout section of the *Dose Response Experiment* is a Spreadsheet section that initially contains sample concentrations populated from the Pipeline Pilot protocol, Calculate Plate Layout that was run within the Plate Details section.

When you click the toolbar item, **Import Instrument File**, you see the Pipeline Pilot Client **Protocol Settings** dialog for the Calculate EC50 Values protocol. The dialog enables choosing the file that contains the raw data that was collected when this experiment was run. The data file is the file, `MicroBeta.txt`.

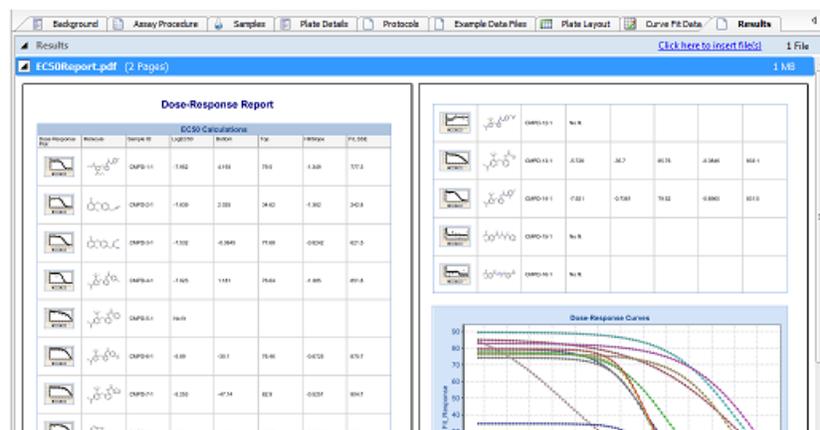
When you click **OK**, you see the **Run Pipeline Protocol** dialog, which disappears after a successful execution is complete. The Calculate EC50 Values protocol imports the sample response data from the instrument file, from the file `MicroBeta.txt` in this example, into the Plate Layout section and calculates the EC50 values. The EC50 data is added to a table section called Curve Fit Data and a File section called Results.

The Plate Layout section displays the concentrations and raw dose-response results for each sample tested:

The Curve Fit Data section displays the Log EC50 and curve parameter values for each sample tested:

Sample ID	Target	Log EC50	Hill Slope	Fit SSE	Bottom	Top	Fit Status	
1	CMPD-1-1	IKK-2	-7.952	-1.349	777.3	4.193	79.5	OK
2	CMPD-2-1	IKK-2	-7.839	-1.382	242.8	2.025	34.62	OK
3	CMPD-3-1	IKK-2	-7.532	-0.8242	621.5	-0.3845	77.68	OK
4	CMPD-4-1	IKK-2	-7.923	-1.385	851.8	1.181	76.64	OK
5	CMPD-5-1	IKK-2						No fit
6	CMPD-6-1	IKK-2	-5.89	-0.6725	875.7	-30.1	76.46	OK
7	CMPD-7-1	IKK-2	-5.253	-0.5251	804.7	-47.14	82.9	OK
8	CMPD-8-1	IKK-2	-5.957	-0.5936	716.8	-14.68	89.57	OK
9	CMPD-9-1	IKK-2	-9.711	-0.4507	1136	-0.1608	95.59	OK
10	CMPD-10-1	IKK-2	-7.922	-1.15	1947	1.132	74.28	OK
11	CMPD-11-1	IKK-2			1136			No fit
12	CMPD-12-1	IKK-2						No fit
13	CMPD-13-1	IKK-2	-5.726	-0.3946	893.1	-35.7	85.76	OK
14	CMPD-14-1	IKK-2	-7.921	-0.8965	831.5	-0.7361	79.52	OK
15	CMPD-15-1	IKK-2						No fit
16	CMPD-16-1	IKK-2						No fit

The Results section displays a Dose-Response report that contains the EC50 calculations and dose response plot for each sample tested.



Synthetic Chemistry Experiment Example

The *Synthetic Chemistry Experiment* template uses the Pipeline Pilot *Calculate Process Mass Intensity (PMI)* protocol. This takes a list of materials as input from a chemical synthesis. The protocol calculates the actual and planned PMI for the products of the reaction and the intermediates, if the reaction is a multi-step synthesis.

The *Synthetic Chemistry Experiment* example is integrated with Pipeline Pilot and the BIOVIA Workbook Process Mass Intensity toolbar option and table section.

When a user clicks the Calculate PMI button, the Calculate PMI protocol extracts data from the BIOVIA Workbook Synthetic Chemistry section, processes the data through a set of calculations, and puts the results into the Process Mass Intensity section.

The example calculates the PMI value for the chemical process involved in a specific reaction. PMI is an indicator of the *greenness* of a chemical process, and it is calculated as the quantity of raw materials input (kg) divided by the quantity of bulk Active Pharmaceutical Ingredient (API) out (kg).

PMI value are used to supplement process-development activities and to help develop less mass-intensive and more productive processes. A low PMI for the process under evaluation indicates a reduced raw material usage, and often corresponds to a better volumetric productivity, and cycle times.

Interactive Overview

If you have template author permission, you can set up a dynamic toolbar item to run an interactive Pipeline Pilot protocol.

The protocol displays a sequence of interactive web pages before returning the data to an experiment.

The Workbook scientist can view a protocol in a Browser dialog, and exchange data with the protocol. In the *Reaction Condition* protocol that comes with the `scitegic_notebook.pib` file, under the Examples folder.

A protocol created for this purpose returns a web page to be hosted on the browser window. Any interaction on that page is handled by the protocol that returns subsequent pages for display and further interaction. A *Finish* event has to be triggered from the protocol that notifies Workbook that the protocol has completed. After the protocol is completed, the final result files are transferred to Workbook.

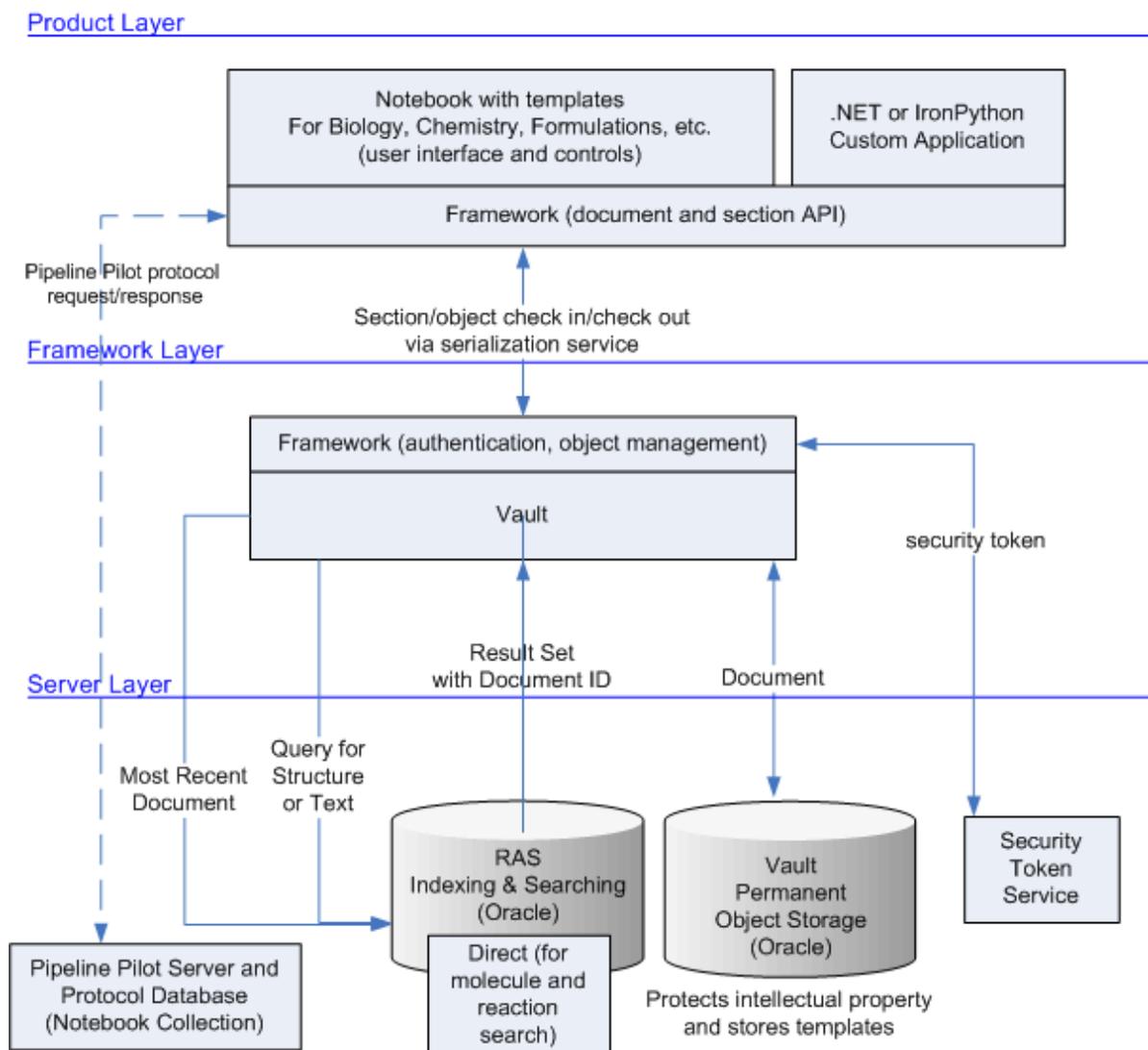
Dynamic Toolbar Item and RunProtocol Permissions

The template author creates a dynamic toolbar item and assigns the `RunProtocol` permission to the toolbar. The permission enables Pipeline Pilot and enables browsing for and attaching a protocol.

All protocols in the Interactive folder are Interactive protocols. The input data and output data are set by the user. When attaching the protocol, the script to execute the protocol is auto-populated.

Integration Architecture

A template author can set up an experiment template to enable processing experiment data with a Pipeline Pilot protocol. The scientists can also import data or reports from the protocol into BIOVIA Workbook experiment sections. The following architectural diagram shows the integration with Pipeline Pilot.



Build Templates that Integrate with Pipeline Pilot

A BIOVIA Workbook template developer can create an experiment template that links to a Pipeline Pilot protocol. A button on the experiment toolbar initiates a command to execute the protocol. The template developer can also use IronPython scripts.

The template developer defines the following:

- Specifies the type of data flows, as well as, the data types the protocol returns to BIOVIA Workbook.
- Specifies the parameter values that are passed to the protocol to control the behavior of the Pipeline

Pilot process.

- Specifies runtime parameters that a scientist can update.

The scientist can export Workbook data as input to the Pipeline Pilot protocol and import data output from the protocol into the Workbook experiment.

Workbook supports exporting data from the following types of experiment sections:

- Table section
 - Analytical Materials
 - Equipment
 - Formulation
 - Formulation Materials
 - Materials
 - Preparations
 - Recipe Materials
 - Sample Analysis Table
 - Synthetic Chemistry

- Forms section

- Spreadsheet section

The integration template can enable a scientist to choose a file as input to the Pipeline Pilot protocol.

An Workbook experiment can import Pipeline Pilot protocol data into the following sections:

- Table section and its derivatives
 - Analytical Materials
 - Equipment
 - Formulation
 - Formulation Materials
 - Materials
 - Preparations
 - Recipe Materials
 - Sample Analysis Table
 - Synthetic Chemistry
- Spreadsheet section
- File section

Create an Experiment Template to Run a Pipeline Pilot Protocol

This example illustrates the integration by allowing BIOVIA Workbook to send data to a Pipeline Pilot protocol and then importing the output of that Pipeline Pilot protocol into Workbook. For this example, a Pipeline Pilot protocol that can take arbitrary, comma-separated values and produce a simple scatter plot graph of the data and return the graph as a PDF. This example links this protocol to an experiment to chart the Workbook data in a graph.

Prerequisites

The Vault Administration Console must set the Workbook client's application permission for PipelinePilot | RunProtocol.

The ProtocolRoot controls the starting point for the selector dialog tree view in the Notebook Explorer:

For more information, see [Pipeline Pilot Protocols Organization](#). In this example, a simple **Property Set Definition** is needed that contains two numeric fields:

- Use the **Property Set** editor to create a new **Property Set Definition** called AnimalInfo.
- Add two fields to the AnimalInfo property set definition, Age and Weight, and set the data type on both of them to Integer.

Configuration steps

To create a template that contains a table section and a file section.

1. In the Notebook Explorer, click **Create > New Blank Template** or select an existing template for the experiment.
2. Add the **AnimalInfo** property set definition to the Table section.
The table should contain two numeric fields, age and weight.
3. On the **View** menu, click **Properties** to activate the table section to facilitate adding the Chart Data button to the table section.
4. In the Section property sheet, under Template, click the ... button for Dynamic Toolbars.
5. Clicking **Add Toolbar** to add a new Dynamic Toolbar to the experiment section.
6. Type a descriptive name for the toolbar such as "Charting".
7. Click **Add Toolbar Item** to add a new Dynamic Toolbar button to the Charting toolbar by .
8. Give the Dynamic Toolbar Button a descriptive name and descriptive display text, such as "Chart Data".
9. Attach the proper application permission to the button. Use the Application Permission that was set up in [Prerequisites](#) PipelinePilot: RunProtocol. Attaching the proper Application Permission enables the Pipeline Pilot protocol option on the button.
10. In the **Pipeline Pilot Protocol** row, click ... to open the dialog for configuring the Pipeline Pilot protocol.
11. Select a Pipeline Pilot protocol to use with the experiment.
For this example, use the **XY Scatter Plot Utility** in the Web Services > Accelrys > Notebook > Examples > General > XY Scatter Plot Utility folder.
12. In the **Pipeline Pilot Protocols** dialog, configure the **Parameters** tab.
For the XY Scatter Plot Utility protocol, the two parameters to configure are X Pproperty and Y property because these parameters tell the protocol which CSV values in the input data are included in the chart. This examples sends data from the Table section to the Pipeline Pilot protocol and the data to chart is Animal Age versus Animal Weight. Table data exported from Workbook uses CSV column headers in the form of PSD Name :: PSD Field. To get the results, for the X property parameter set the value to AnimalInfo::Age, and for the Y property set the value to AnimalInfo::Weight.
13. On the **Input Data** tab, specify the Table section as the source of the input to the protocol.
14. On the **Output Data** tab, specify the File section as the recipient of the output from the protocol.

15. Click **OK**.

This saves the integration specifications into the Chart Data Dynamic Toolbar Item.

16. Look at the **Script** property on the Dynamic Toolbar Item. The Script is set to a default IronPython script that calls the Pipeline Pilot protocol.

You do not need to edit the default script. The content of the default script is similar to the following:

```
import clr import sys clr.AddReference('System.Windows.Forms')
from System.Windows.Forms import (DialogResult)
from Symyx.Framework.PipelinePilot import ProtocolProxy
from Symyx.Framework.PipelinePilot import PipelinePilotPermissionHandler
from Symyx.Framework.Properties.ImportExport import ImportExportFormats
from Symyx.Notebook.PipelinePilot import PipelinePilotHelper

# -----
# Main Method
# -----
def MainMethod():
    global protocol, editor, application_permission
    if not PipelinePilotHelper.ExportProtocolData(protocol,
editor.Document):
        return
    server = PipelinePilotPermissionHandler.GetPipelinePilotServerProxy
(application_permission)
    if not PipelinePilotHelper.RunProtocol(server, protocol,
intervalMilliseconds, timeoutSeconds):
        return
    PipelinePilotHelper.ImportProtocolData(protocol, editor.Document)
    PipelinePilotHelper.DeleteProtocolTempFiles(protocol)

# -----
# Main part of the script
# Just invokes the MainMethod
# -----
MainMethod()
```

Use cases for customizing the script include:

- Pre- or post-processing data from an external laboratory instrument
- Working with data from an SDfile

Test the Capability to Run the Protocol

To test the capability to run the protocol, get a scientist to create a new experiment from the experiment template set up in the Configuration steps. In the experiment template, have the scientist navigate to the Table section's toolbar, called Chart Data.

Ask the scientist to enter data for the animal's age and weight and then click the **Chart Data** button to run the Pipeline Pilot protocol.

If the **Run Pipeline Pilot Protocol** dialog displays the progress of the protocol, you can conclude that the end user has the capability to run a Pipeline Pilot protocol.

Enable Changes to Runtime Values

When you create a Dynamic Toolbar item that is used in conjunction with a Pipeline Pilot protocol, the information you enter into the **Protocol Selector** dialog, that is, the values of parameters, input data, and output data, is stored with the Dynamic Toolbar item when the document template is saved.

When the scientist selects the Dynamic Toolbar item, the values are passed to the Pipeline Pilot protocol. In the case of a file on disk, the name of the file on disk is stored, not the content of that file. When the scientist selects the Dynamic Toolbar item, that file is read and its content is passed to the protocol.

The `Symyx.Notebook.PipelinePilot.PipelinePilotHelper.CheckProtocolData` method enables specifying the Pipeline Pilot protocol to run to allow the scientist to change the values of parameters, input data, and output data. This method checks at runtime whether any values were specified as changeable and, if so, presents a dialog that allows the scientist to supply values.

Parameters can change the behavior of the protocol. Specify the source of the raw data for the protocol, or specify the type of file to output.

Input data can come from a file on disk instead of a document section. For an example that takes input from a file on disk, see the [In Vivo Experiment example](#). An IronPython script or or .NET developer who selects the protocol to run, can specify whether:

- One or more parameter values can change at runtime.
- Data input into the protocol can come from a document section or from a file on disk.
- The sections into which protocol-generated data are added to the document are changeable at runtime.

If the data come from a document section, do not mark the data as `changeable at runtime`. If the data come from a file on disk, you can mark the data as `changeable at runtime`.

The `changeable` flag allows the end user to make changes at runtime. If the Pipeline Pilot protocol author for the XY Scatter Plot Utility protocol added a new parameter called *NumberOfPointsToGraph*. You could set a default parameter value to 10 and mark this parameter as `changeable`. When the scientist runs the protocol in an experiment created from the template, the scientist sees a runtime parameter dialog that allows the scientist to increase or decrease the default number of points to graph.

Script Modifications to Customize the Integration

The following IronPython script allows Workbook to make use of Pipeline Pilot protocols.

To access the script:

- In the Notebook Explorer, navigate to the **Dynamic Toolbar Editor** dialog and click the ... button.

```
import clr
import sys
clr.AddReference('System.Windows.Forms') from System.Windows.Forms
import (DialogResult) from Symyx.Framework.PipelinePilot
import ProtocolProxy from Symyx.Framework.PipelinePilot
import PipelinePilotPermissionHandler from Symyx.Framework.Properties.ImportExport
import ImportExportFormats from Symyx.Notebook.PipelinePilot
import PipelinePilotHelper
# Main Method
def MainMethod(): global protocol, editor, application_permission
protocolRef =
clr.Reference[ProtocolProxy](protocol)
    if not PipelinePilotHelper.CheckProtocolData(sender.Text, protocolRef, editor):
        return protocol = protocolRef.Value
        if not PipelinePilotHelper.ExportProtocolData(sender.Text, protocol,
editor.Document):
            return server =PipelinePilotPermissionHandler.GetPipelinePilotServerProxy
(application_permission)
            if not PipelinePilotHelper.RunProtocol(server, protocol, intervalMilliseconds,
timeoutSeconds):
                return PipelinePilotHelper.ImportProtocolData(sender.Text, protocol, editor.Document)
                PipelinePilotHelper.DeleteProtocolTempFiles(protocol)
# -----
# Main part of the script# Just invokes the MainMethod
# -----
MainMethod()
```

Processing Data for Export to Pipeline Pilot

To process data before exporting to a Pipeline Pilot protocol, use the `PipelinePilotHelper.ExportProtocolData(sender.Text, protocol, editor.Document)` function:

```
if not PipelinePilotHelper.ExportProtocolData(sender.Text, protocol,
editor.Document):
    return
```

The `ExportProtocolData` method extracts data from the sections in the experiment and writes the data to temporary files.

The protocol has a `FileInputs` property.

Calling the `FileInputs` property returns an enumeration of `Symyx.Framework.PipelinePilot.ProtocolFileInput` that contains information about the data input such as

- Whether the data comes from a file on disk or a document section in the `DataSource` property.
- The section name in the `SectionName` property.
- The location of the file in the `FilePath` property.
- The path to the source file.
- The destination file where data is written by the `PipelinePilotHelper.ExportProtocolData` method when the data comes from a document section.

- The `FilePath` value is not generated until the `PipelinePilotHelper.ExportProtocolData` method.

Accessing data in an external system

The Protocol Selector user interface provides access to data contained within a file on disk or a document section. However, you can modify the script to provide access to data from an external source such as an Oracle database. To access the data, before the call to `PipelinePilotHelper.CheckProtocolData`, make a call to the external data source, or issue an Oracle SQL statement. Here is a possible scenario:

1. When setting up the protocol to run, select **File on Disk** as the data source.
2. You can select that the file path is changeable at runtime. If the path is `changeable at runtime`, you do not have to specify a file path at design time.
3. In the script, set the value to `not changeable at runtime`, and also supply the file that contains the data accessed from the external data source. You could specify that the file path is `not changeable at runtime` and have the script change the `FilePath` that was selected in the **Protocol Selector** user interface.
4. Before the call to `PipelinePilotHelper.CheckProtocolData`, enter the code that accesses the external data source, and writes the accessed data to a file on disk.
5. Set the appropriate `ProtocolFileInput.FilePath` property to the path to the file on disk.
6. To ensure that the `FilePath` is *not changeable at runtime*, set the `ProtocolFileInput.ChangeableAtRuntime` property to `False`.

Modifying Parameter Values

To modify parameter values using the script, insert your code before the call to `PipelinePilotHelper.CheckProtocolData`. The protocol has a `Parameters` property that returns an enumerable of `Symyx.Framework.PipelinePilot.ParameterInfoProxy`. A `ParameterInfoProxy` has a `Values` property that can be used to modify the value of a parameter. A parameter can contain multiple values (all strings), and the `Values` property returns an array of strings.

Post-processing Pipeline Pilot Data

You can post-process the data generated by the Pipeline Pilot protocol before or after the data is added to the Workbook experiment.

To post-process before importing to the Workbook experiment, the location in the script is after the following code:

```
if not PipelinePilotHelper.RunProtocol (server, protocol, intervalMilliseconds,
timeoutSeconds):
return
```

At this point, the data generated by the protocol resides on the local client computer running the Workbook client because `PipelinePilotHelper.ImportProtocolData(sender.Text, protocol, editor.Document)` has not yet been called.

To post-process after importing to the Workbook experiment, the location in the script is after the call to the `ImportProtocolData` method. You can have a combination of the following:

- Non-post-processed data go into the originally specified section after the call to `PipelinePilotHelper.ImportProtocolData`.
 - Post-processed data to go into another section in the document which would be done with scripting
- The protocol in the script, the instance of `Symyx.Framework.PipelinePilot.ProtocolProxy`, has a `FileOutputs` property. This property returns an enumerable of `Symyx.Framework.PipelinePilot.ProtocolFileOutput` that contains information about the data returned from running the protocol such as:
- The location of the file on the client computer into which the data was written, the `TempFilePath` property.
 - The name of the section into which the data is to be written, the `SectionName` property.

.NET APIs and IronPython Script

You can modify the integration by using an IronPython script. The IronPython code makes use of the .NET application programming interfaces that support the integration of BIOVIA Workbook and Pipeline Pilot protocols.

In the `Symyx.Framework.PipelinePilot` namespace, the following classes:

- `ProtocolProxy`
- `ParameterInfoProxy`
- `ProtocolFileInput`
- `ProtocolFileOutput`
- `PipelinePilotPermissionHandler.GetPipelinePilotServerProxy` method

The `Symyx.PipelinePilot` namespace is available in the `Symyx.Framework`.

The `ProtocolRunner` class is in the `Symyx.Framework.Controls.PipelinePilot` namespace.

In the `Symyx.Framework.Properties.ImportExport` namespace contains the following enums:

- `ImportExportFormats`
- `ImportBehavior`
- `ExportBehavior`

The `PipelinePilotHelper` class is in the `Symyx.Notebook.PipelinePilot` namespace.

An experiment's Document section can import data and implement the `Symyx.Framework.Properties.ImportExport.IImportable` interface. Document sections can export data and implement the `Symyx.Framework.Properties.ImportExport.IExportable` interface.

For details, see the API Reference for BIOVIA Workbook and Framework.

Authoring Protocols Overview

You can use BIOVIA Pipeline Pilot to perform calculations, analyze, and generate reports on large quantities of scientific data. You can find more information about authoring Pipeline Pilot protocols in the following documentation:

The following considerations are specific to protocols that work with BIOVIA Workbook.

- [License Requirements for Workbook Specific Pipeline Pilot protocols](#)
- [Development Process for Pipeline Pilot Protocols](#)
- [Pipeline Pilot Protocols Organization](#)
- [Pipeline Pilot Protocols Guidelines](#)

License Requirements for Workbook Specific Pipeline Pilot protocols

BIOVIA Workbook is shipped with a working Pipeline Pilot Server that contains one production Pipeline Pilot protocol and Pipeline Pilot protocol examples.

To author Pipeline Pilot protocols, you must obtain a license for the Pipeline Pilot Client and any collections that you want to use to process your data.

Development Process for Pipeline Pilot Protocols

Your organization has a production Pipeline Pilot that scientists use on a daily basis to record the results of their experiments. The organization might have a test environment for trying out new versions of the software, new experiment templates, or new configurations, in a safe *sandbox* without disrupting the production system.

You can install a Pipeline Pilot Server with both the production and test BIOVIA Vault Server environments. BIOVIA recommends that you do all Pipeline Pilot protocol development on the test Pipeline Pilot Server to enable developing the experiment template and Pipeline Pilot protocol in the test environment before promoting the functionality to the production system.

When an experiment template and its attached protocol have been validated in the test environment, the protocol developer can export the protocol as an XML file and imported into the production system. The developer can create an experiment template on the production system that connects to the imported Pipeline Pilot protocol.

Pipeline Pilot Protocols Organization

Set the Pipeline Pilot RunProtocol Application Permission using the Vault Administration Console to enable the integration point between BIOVIA Workbook and Pipeline Pilot. The endpoint configuration specifies the Pipeline Pilot Server endpoint used by the scientists when running protocols from their BIOVIA Workbook experiments.

The ProtocolRoot configuration specifies the subset of Pipeline Pilot protocols that are visible from the **Pipeline Pilot Protocol Selector** dialog. The ProtocolRoot defaults to the value of Protocols/web_services/workbook/Experiments.

If your company develops customized Pipeline Pilot protocols, BIOVIA recommends that you save your protocols to

Protocols/web Services/workbook/Extensions to enable viewing both the BIOVIA and the custom protocols in the Pipeline Pilot Client Protocol Selector dialog without the need to change the ProtocolRoot value.

Pipeline Pilot Protocols Guidelines

Notebook Readers and Writers

The Reader components in the Notebook collection encapsulate the complexity of reading Workbook data formats.

Notebook Form Reader

You can send data from the Workbook form sections to Pipeline Pilot as input to a protocol. The *Notebook Form Reader* component reads the data into a protocol, and can apply the form entry values to the protocol parameters.

The component outputs one record for each form that it reads. The properties in each form correspond to the form element names defined in the Workbook form, and the value of each property is the data entered for that form element.

A **FormTitle** property defines the title assigned to each of the Workbook forms read in to the protocol. Use the form title to distinguish elements of the same name contain in multiple forms.

The *Notebook Form Reader* component can assign form values to the protocol parameters. When this is set to True, form values for elements that match the name of a protocol parameter are copied to the matching protocol parameter. For example, if the Workbook form has an element named ExperimentID, and the protocol is given a parameter also named *ExperimentID*, you can use this option to automatically set the form value for ExperimentID to the protocol parameter. This causes **ExperimentID** to appear as a global property for the protocol.

Notebook Table Reader

You can send data from the Workbook table sections to Pipeline Pilot as input to a protocol. The *Notebook Table Reader* component reads the data in to a protocol.

If the table data contains a hierarchical organization of rows, these are collapsed into a single hierarchical data record in Pipeline Pilot.

Use the *Maximum* setting to limit the number of records that are fed out of the pass port of the component. Hierarchical data does not usually to reach the specified *Maximum* value. *Maximum* is the greatest number of records generated by the reader component.

For example, to generate 100 hierarchical records might take 500 table rows from Workbook: 100 master rows and 400 child rows. This number refers the number of records output from the reader. The *Notebook Table Reader* read as many rows from the Workbook-supplied file to reach that maximum, unless the reader reaches the end-of-file (EOF).

When molecular structure and reaction data is detected in the input table records, the generated Pipeline Pilot data record or hierarchical node is mapped to the appropriate chemical type.

Where appropriate, each property in the generated data record reflects the data type of the corresponding field in the Workbook table.

The name of each property is constructed from the name of the *Property Set Definition (PSD)* and the name of the field for the corresponding data item in the Workbook. These two names are separated by a double colon (::) in the Pipeline Pilot property name. For example, Material::Name.

Notebook Table Writer

You can import data from a Pipeline Pilot protocol into a Workbook table section. The Table section requires a specific CSV format for the data.

The *Notebook Table Writer* component writes a record table based on the content of the data records that flows into the component.

The output properties use the format: [data-type]field-name. You can specify the data type if there is a corresponding property on the data record named field-name datatype. For a property read by the *Notebook Table Reader*, this property is created automatically.

For a data record that is a molecule or reaction data record, a property is created with the encoded representation. The name of the field is taken from the *Encoded Representation Field Name* parameter.

When a hierarchical data record is encountered, the *Notebook Table Writer* component writes each detail record row for each child node on the input data record. Each row repeats the columns of the master record, along with the columns of the detail record. Each row also pre-pends columns named Level0 and Level1 to indicate which master and detail records are represented in that row. The output table uses ascending order (1, 2, 3 ...) by the Level0 and Level1 column values.

Table Section CSV Format

Pipeline Pilot protocol outputs a comma separated variable (CSV) format with a comma as the field separator.

There is a required header line and a line for each row of data to import, with one comma delimited field for each column.

The data headers have three required components in the format:

```
[TypeName]PropertySetKey::PropertyKey
```

For example,

```
[String]Material::Name
```

If the data contains the comma character, the string is *protected* by surrounding the data in double quotes.

For Master and Detail files, the first two columns in the header use the following form:

```
Level0,Level1
```

The first two columns in the data represent the master/detail rows for the data. For example:

```
0,0,data1,sub1
0,1,data1,sub2
1,0,data2,sub1
1,1,data2,sub2
1,2,data2,sub3
2,1,data3,sub1
```

For example:

```
0,0,Aspirin,Binder
0,1,Aspirin,ActiveIngredient
1,0,Bufferin,Binder
1,1,Bufferin,ActiveIngredient
1,2,Bufferin,Caffeine
```

The headers of the master detail case add another portion to the header string:

```
[level][TypeName]PropertySetKey::PropertyKey For example: [0]  
[String]Material::Name, [1][String]Material::Name
```

Configure the Protocol

To integrate a protocol with BIOVIA Workbook, follow these conventions:

- Protocol Level Parameters

Set the visibility of any parameter in the protocol to show in the Workbook integration point. For example, promote the **Protocol Selector** Dialog to the Protocol level. Only template developers and users can set protocol-level parameters at runtime.

- Protocol Output Files

Any protocol output file that you want to import into Workbook needs to be declared on the protocol's Web Service tab as a Results File.

- Parameters as Array Values

The integration with Workbook only supports parameters array values for String type parameters. If you need to pass multiple values to a protocol for a parameter of another type, you must create multiple parameters for each value needed in the protocol.

Troubleshooting

Developing Pipeline Pilot protocols for BIOVIA Workbook is an advanced task that requires expertise in both the Workbook property sets and Pipeline Pilot protocol development.

Data import issues

Things can go wrong when trying to import data. Enable logging in Workbook client to review log statements that might indicate if there were data format errors or other problems with the import. You can enable logging by removing the comment from the `configuration/log4net/root/appender` note found in the `Symyx.Notebook.Application.exe.config` file.

Format inconsistencies

When your Pipeline Pilot protocol takes uses the Workbook table data or forms data, export the section data as a file so that you can look at the data and understand its format. You can use the Dynamic Toolbar to add a button to the section that is used as input. You can then add a Python script to the button to export the data to a file. Using this approach provides a means to see the exported file and verify that the data is in the same format that Pipeline Pilot uses.

The Python script to export data from a Workbook forms section can look similar to the following:

```
import clr  
import sys  
from System import (Exception, StringComparison) from  
System.Collections.Generic import Dictionary from System.IO import (File,  
FileStream, Path)  
from Symyx.Framework.Properties.ImportExport import ImportExportFormats  
from System.Windows.Forms import (DialogResult, MessageBox,  
MessageBoxButtons, MessageBoxIcon)  
outputPath = "c:\\temp\\forms output.xml"  
  
try:  
fs = File.Create(outputPath)
```

```
owner.Export(fs, ImportExportFormats.XML, None)
except Exception, e1:
    errMsg = 'Error creating file ' + outputPath + ': ' + e1.Message
    MessageBox.Show(errMsg, 'Export error')

finally:
    fs.Close()
```

The Python script to export data from a Workbook table section can look similar to the following:

```
import clr import sys
from System import (Exception, StringComparison) from
System.Collections.Generic import Dictionary from System.IO import (File,
FileStream, Path)
from Symyx.Framework.Properties.ImportExport import ImportExportFormats
from System.Windows.Forms import (DialogResult, MessageBox,
MessageBoxButtons, MessageBoxIcon)
from Symyx.Framework.Properties.ImportExport import ExportBehavior
outputPath = "c:\\temp\\table output.csv"

try:
    fs = File.Create(outputPath)
    owner.Export(fs, ImportExportFormats.CSV, None, ExportBehavior.ADDROWID)
except Exception, e1:
    errMsg = 'Error creating file ' + outputPath + ': ' + e1.Message
    MessageBox.Show(errMsg, 'Export error')

finally:
    fs.Close()
```

Protocol issues

After you have exported the data files from BIOVIA Workbook, upload the data files to the Pipeline Pilot Server. While you are developing the protocols, use the data files local to the Pipeline Pilot Server as input to the protocol. You can then use the Watch and Check Point features of the Pipeline Pilot Client to debug the protocol.

Salt stripping problems

The salt stripping function Remove Salts (in Enumeration Products section of Parallel Chemistry experiment) fails when the **ConfigFilePath** and **VaultEndPoint** global properties are not set in the Pipeline Pilot Admin Portal.

To set salt stripping global properties:

1. Open the Pipeline Pilot Admin Portal, for example:
`http://<PPserver>:<port>/admin`
2. Select the **Setup | Global Properties** page from the Admin explorer.
3. Choose **BIOVIA/Workbook** from the Package list.
4. Select the **ConfigFilePath** property and set its value as appropriate, this is the full path on the Pipeline Pilot Server to the folder containing the `Salts.sd` file required by the Remove Salts protocol. Click **Save**.
5. Select the **VaultEndPoint** property and set its value as appropriate, this is the fully qualified name of your Vault server. Click **Save**.

Limitations with Previous Versions

Salts.sd

It is possible to modify the list of salts in the `salts.sd` file. For example, you might want to add additional salts. If you are the Workbook administrator and you have customized the `salts.sd` file in a Workbook version from a previous release, new installations overwrite the existing `salts.sd` file. You can make a backup of the existing `salts.sd` file from your current installation and after installing a new version, replace the `salts.sd` file with the backup copy. Make sure that the name remains `salts.sd`.

Limitations for integration with Pipeline Pilot

- You can only data from the Workbook forms sections, table sections, and spreadsheet sections to a Pipeline Pilot protocol.
- You can only import data returned from a Pipeline Pilot protocol in to the Workbook table sections, spreadsheet sections, and file sections.
- Some Pipeline Pilot protocols can generate interactive web pages that are designed to display in a web browser. Notebook 6.6 cannot display such web pages.
- You should limit quantity calculations to simple arithmetic and scalar transformations that avoid unit type conversions.

The data types that Workbook uses are often quite complex. For example, a quantity captured in Workbook consists of a number, a number of significant digits, and a unit. There is no built-in Pipeline Pilot capability to handle significant figures and units. Complex calculations that involve the conversion of a quantity from one unit type to another and the handling of significant digits are the responsibility of the protocol developer.