



WHAT'S NEW IN QLIKVIEW EXPRESSOR 3.9

A QlikView Technical Brief

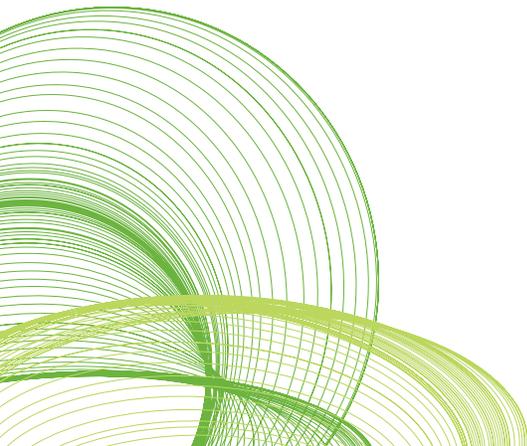
March 2013

qlikview.com



Table of Contents

What's New in QlikView Expressor 3.9?	3
Semantic Conformance Mapping	3
Multi-transform Operator	4
Metadata Analysis Search and Usage	5
Extensions SDK	8
What is a QlikView Expressor Extension?	8
The Extensions SDK	8
Resources	11



What's New in QlikView Expressor 3.9?

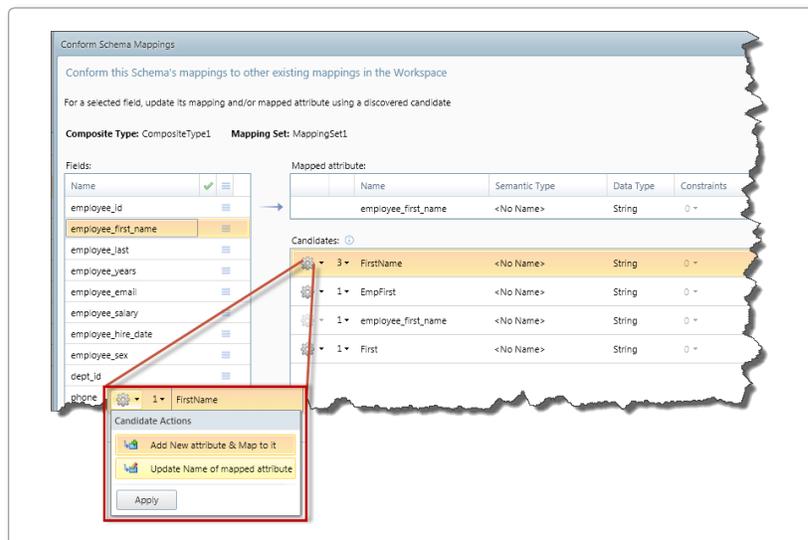
We are excited to announce a host of new features that will improve semantic conformance, introduce metadata analysis, increase productivity and allow you to build your own components when using QlikView Expressor Studio. We are always working to make QlikView Expressor more functional and intuitive, and we're listening. If you have anything you'd like to tell us, don't hesitate to let us know.

SEMANTIC CONFORMANCE MAPPING

Data attributes used by QlikView should be easy to recognize and prepared with consistency. Determining whether “account_num” is used in the same manner as “customer_num” or whether “Profit Margin” is similar to “Gain”, can lead to confusion and delay in decision making. As applications grow rapidly, management of data and how it's prepared can become an afterthought. QlikView Expressor has added a new feature that ensures data attributes remain consistent across multiple QlikView Expressor projects. The Conformance Mapping interface enables consistent naming of attributes representing a given data element. It reduces the proliferation of unnecessary variations that can cause inconsistent results in data processing. Conformance Mapping makes it simple to unify the attributes mapped to fields in a Schema. When an attribute shares similar properties with other attributes in the Workspace, it can be made to conform to one of the other attributes. QlikView Expressor Studio supports this in the following manner:

- For a given field in a schema, the Studio shows other attributes in the workspace that are mapped to fields of the same name known as Candidates.
- The user may then update the given field's mapped-attribute, to conform to one or more already existing in the workspace.

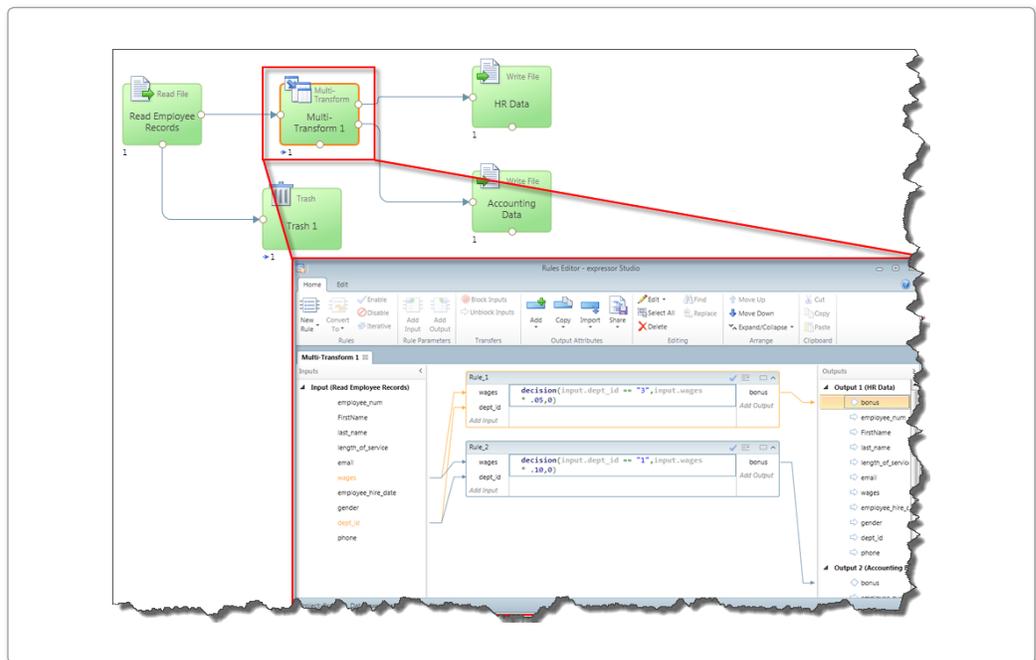
Figure 1: Conformance Mapper, displaying list of possible candidates for the employee_first_name field



MULTI-TURNFORM OPERATOR

The Multi-transform operator improves the overall efficiency and management of large dataflows that may require finer control on how data is transformed and where it resides. It enables data to be transformed and directed without the need for additional Filter operators. It is particularly useful when processing transformation logic that needs to output records having different structures to different destinations. This is often the case when the transform logic takes in one input record, uses that record to identify more content (of various forms) and uses an iterator¹ to produce multiple output records from that content.

Figure 2: Multi-transform operator with multiple output ports and expression rules



¹ iterator is a function or object that enables a program to traverse a set of data

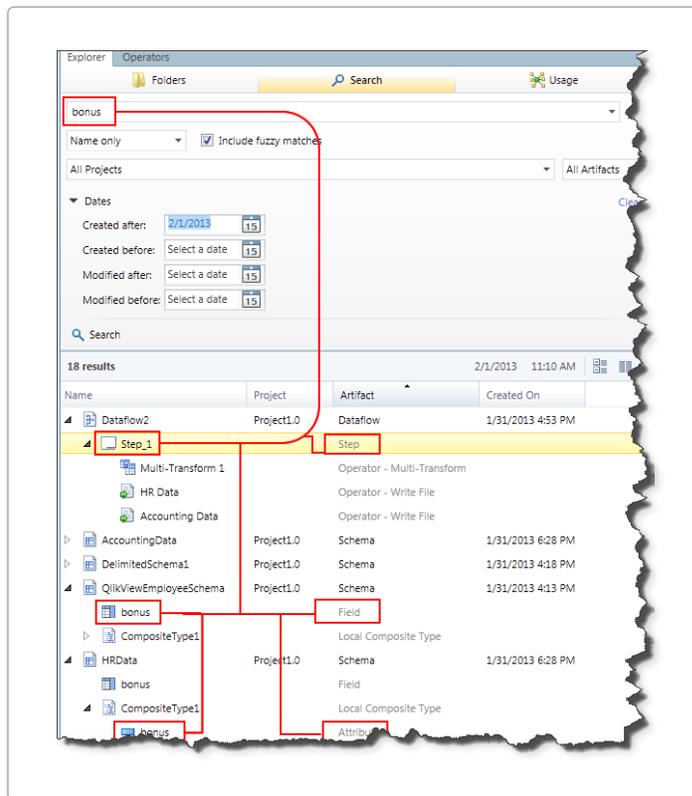
METADATA ANALYSIS SEARCH AND USAGE

You cannot manage what you can't find. Understanding where data elements originate, how they are processed and where they end up reduces the many risks associated with managing data, such as its security, privacy and sensitivity. Managing these risks is necessary, but there's one problem: in order to achieve compliance companies must be able to reveal where data comes from, where it flows to, and how it's transformed as it travels through the enterprise. This is where tools that provide "impact analysis" come in handy; being able to analyze where data is, and how it flows so you can manage and secure it appropriately as it moves across the Enterprise.

The Metadata Analysis Search and Usage interface helps locate QlikView Expressor artifacts (dataflows, schemas, types, connections, attributes, fields, etc.) within workspaces, projects and libraries that are used to manage and prepare data for QlikView and other data targets. It achieves this using the following two methods, Search and Usage.

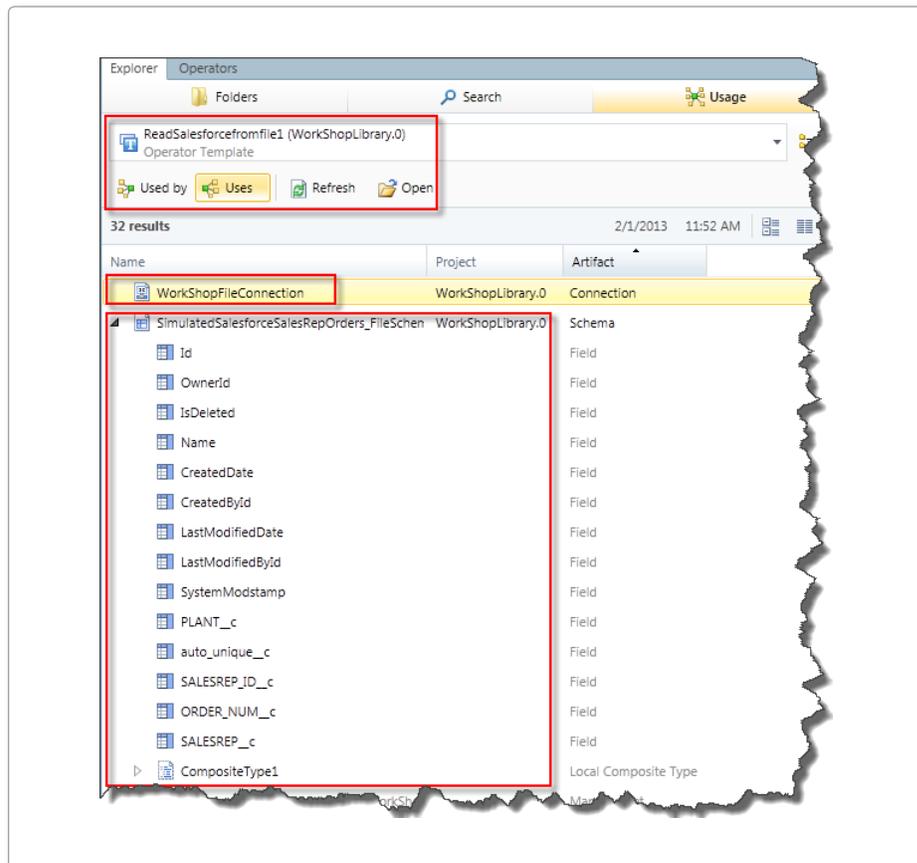
- **Search** – supports a variety of user input parameters, including dates and fuzzy matching, to help you locate various artifacts across the workspace.

Figure 3: Search facility locating defined attribute "bonus"



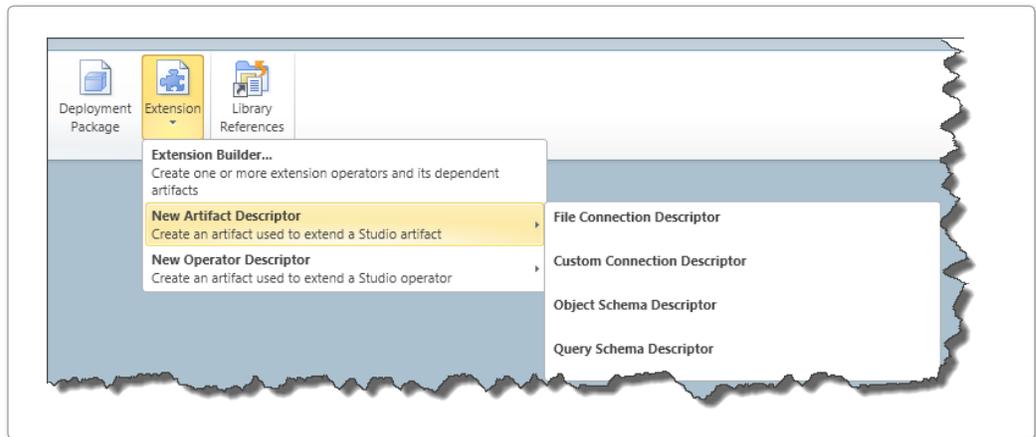
- **Usage** – supports the selection and searching of artifacts by “Uses” or “Used by”
 - **Uses** – displays all artifacts that the *selected* artifact uses
 - i.e. a schema using a type
 - i.e. an operator template using a connection and a Schema

Figure 4: Usage facility locating where and what is using the ReadSalesForcefromFile Template



- **Used by** – displays all artifacts that the *selected* artifact is *used by*

Figure 5: Usage facility locating where or what the MSSQLServerConnection is used by



EXTENSIONS SDK

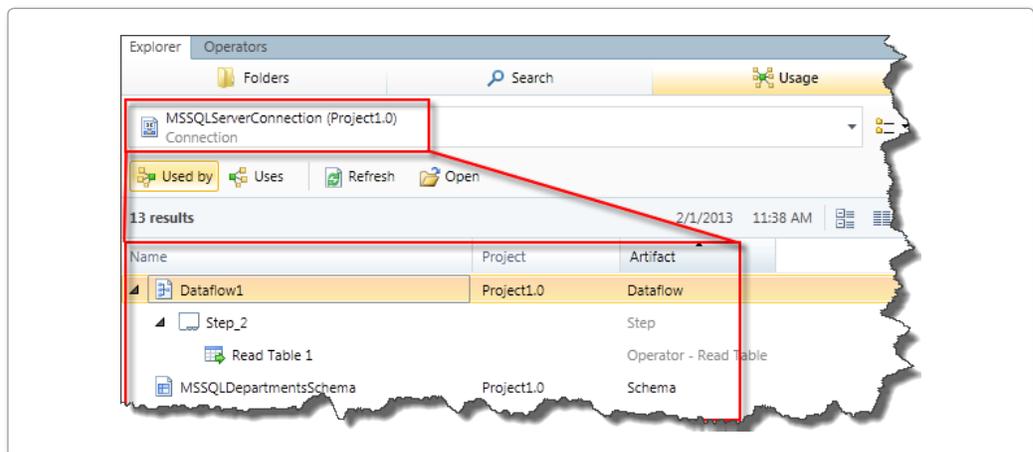
What is a QlikView Expressor Extension?

A QlikView Expressor Extension is a plug-in (or add-on) that adds new operators and metadata artifacts to those included in the core Expressor product. Those capabilities can provide access to data from sources that are not accessible through the standard operators and artifact types such as Read and Write Table, Database and File Connection, and Table and Delimited Schema. Extensions can also be designed to perform transformations on data that cannot be performed by the standard QlikView Expressor operators.

The Extensions SDK

The Extension SDK will enable a set of toolkits and wizards that provide a framework with common functions for developing Extensions for QlikView Expressor. With Extensions, developers can provide clients with tools to integrate data from specialized sources and perform specialized transformations on data. An Extension's operators, Connections and Schema integrate into the Studio interface and seamlessly work in conjunction with other Standard, Enterprise, and Extension artifacts and operators.

Figure 6: Extension Builder with Descriptor Wizards to create custom operators and artifacts



The Extensions SDK provides a set of compilation, utility, and data type conversion functions using built-in Datascript Modules (DSM). DSMs are the standard QlikView Expressor scripting module for Datascript which is based on the Lua open-source scripting language. Extensions build on these common functions add functionality to the QlikView Expressor Studio and support custom data sources and specialized transformations.

Extensions developed with the SDK can be installed easily into an existing QlikView Expressor software installation. All Extensions are installed and activated through the Manage Extensions option on the Studio menu.

Figure 7: Extension created, packaged and then enabled by the QlikView Expressor Extensions Manager

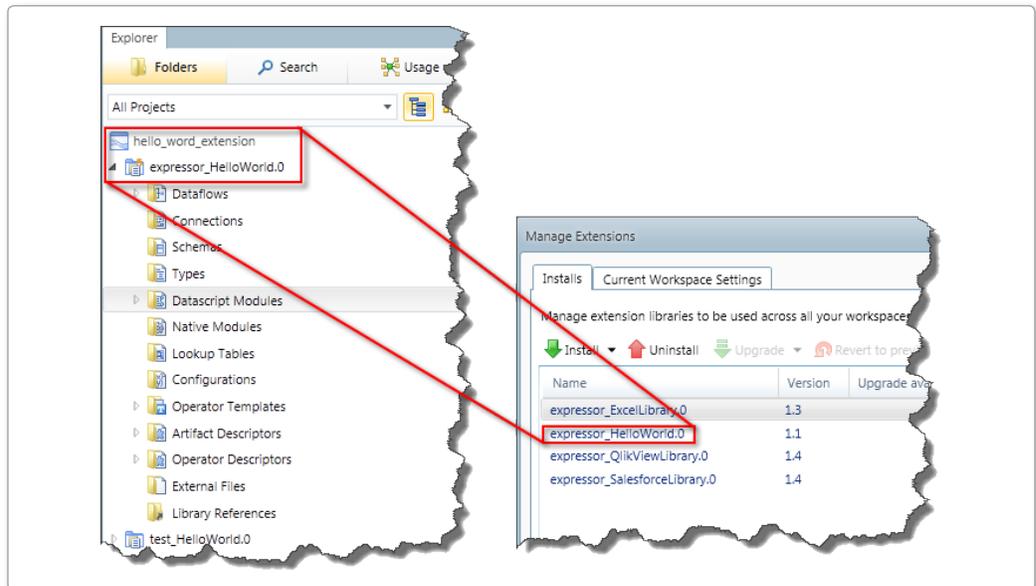
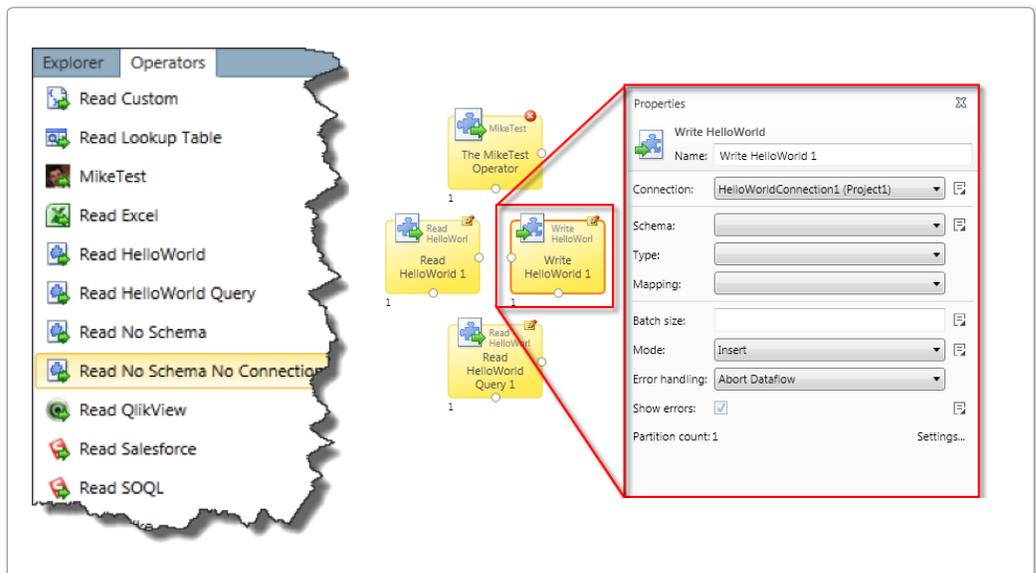


Figure 8: Operators list with custom built operators, dataflow using those operators and custom properties



Resources

For detailed information on these features and much more please review the latest documentation found on the QlikCommunity or installed with QlikView Expressor Desktop.

Documentation

<http://community.qlikview.com/docs/DOC-3325>

Forum

http://community.qlikview.com/community/qlikview_expressor

QlikView Expressor Product Page

<http://www.qlikview.com/us/explore/products/expressor>