



640 West California Avenue Suite 210
Sunnyvale, CA 94086
1-488-992-7100

MRD for Schema Views

Approval

<i>Department</i>	<i>Name</i>	<i>Date</i>
Development		
Marketing		
Sales		
Support		

Modification History

<i>Version</i>	<i>Date</i>	<i>Author</i>	<i>Comments</i>
1.0	3-13-2006	Lenny Hoffman	Initial Draft

Feature Requirements

Description of the Problem

Application designers that need to extend to their end user's ad-hock query capability often need to present class and member's with different names and/or designate classes and members as not visible to user defined queries.

This was a requirement for the custom query engine built by me (Lenny Hoffman) for SAIC. Their problem was that they (as developers) used different names for things than their end users did and they had detail in their schema that end users would not understand and for encapsulation purposes needs to be hidden.

Not only is it common that a developer and a customer use different names for the same thing, but different customers for a given application may use different names, thus the requirement is to have multiple views of schema, not just a single alternative view.

The usefulness of having multiple views goes beyond end user query support; it can also be useful to support multiple development groups working against the same FD. Consider the case that there is a primary group responsible for creating a schema robust enough for the entire enterprise, but individuals from other groups that are best for adding or consuming data from the overall FD would have their task simplified by being presented with a schema view that is a subset of the overall enterprise FD schema and that uses naming familiar to them.

Description of the requested feature

The feature is to be able to add a named schema view to the schema. There is one base schema, which is the actual schema entered into the FD. A named schema view adapts the base schema for a specific user and provides the ability to rename and hide classes and/or members.

It will be possible to add schema view, give it a name, select classes and members from the base schema to be included in the view and give new names to items.

It shall then be possible to set the current schema view to a view other than the base schema, or return the schema view to the base schema.

Part of an optional feature or does it require another feature? If so, which one?

This is an enhancement to the Schema Model feature. The Enhanced Object Qualification feature and the Unified Language Binding feature will then automatically benefit from the enhancement, as they use they use the Object Model feature that will base its class and member lookups on the current schema view.

How is this problem being solved now, and why isn't that acceptable?

It is not being solved now by Objectivity. As mentioned, some customers (e.g. SAIC) have solved the problem by creating their own custom query engine, but we don't want to force customers to have to do that.

Which languages must support this capability?

All.

Which platforms must be supported?

All.

Do any competitors already have this capability?

Yes. SQL Server just introduced schema views into their latest version. They presented it at a Microsoft product launch conference I was at recently and it received much positive interest from the conference participants.

Customers who require this capability:

Those extending ad-hock query to their customers. Those integrating data from multiple sources and delegating import/export duties to constituents.

Revenue at risk or which could be won:

We are risking revenue from the new crop of customers that are looking to us as a more effective and scalable alternative to the relational DB's they

are using now but have found to not be scalable. These customers take advantage of our support for more complex data structures, but need to trim them down to be easily digestible by their customers.

When is this required?

ASAP.

Review

Feature Sizing

<i>Efforts</i>	<i>Size</i>
Development	
QA	
Documentation	

Scheduled for Objectivity Release

Assigned Engineering Group