Objectivity

301B East Evelyn Avenue
Mountain View, CA 94041
1-650-254-7100

MRD for Objectivity Plugin Framework (OPF)

Approval

Department	*Name*	*Date*
Development		
Marketing		
Sales		
Support		

Modification History

Version	*Date*	*Author*	*Comments*
1.0	3/15/2007	Lenny Hoffman	Initial Draft

Feature Requirements

Description of the Problem

Many features are in progress or have been proposed that have an element of extensibility and without a common plugin framework each could provide its extensibility in a different way and would not benefit from a more comprehensive and unified approach.

Description of the requested feature

First some definitions:

Extension point: The part of a feature that can be extended.

Plugin: The extending component.

Plugin Specification: A description of the plugin, which includes the extension point it corresponds to and its capabilities.

Capability: An optional part of a plugin specification that indicates that the plugin is capable of some level of functionality.

Extension point interface: The base class or interface that plugin implementations derive from or implement.

Extension point evolution: Changes made to an extension point interface or to the definition of the plugin's role.

The requested feature is a framework for extension points and plugin contributions that is robust enough to satisfy the needs of any of our current and future feature extension needs.

Goals:

1. Make it easy for us to expose extension points to our customers.
2. Make it easy for our customers to extend our product.

3. Support extension point evolution.

Strategies:

1. Provide a single mechanism for customers to use when extending our product, no matter what that extension is doing.
    a. G1 – We don't have to reinvent the wheel each time we want to add another extension point.
    b. G2 – Users don't have to learn a new way to extend our product each time they use a different extension point.
2. Separate plugin specification data from plugins.
    a. G2 – This makes it easier for users to describe their plugins by not making them do so in code.
    b. G2 – This makes it easier to change the description of plug-ins, as doing so does not require that they be rebuilt.
3. Allow plugin specifications to specify specific capabilities.
    a. G1, G3 – This makes it easier to handle behavior differences not reflected in extension point interface API by not requiring that new API be added just for that purpose.
4. Have a common plugin specification mechanism that is independent of the language plugins are implemented in.
    a. G2 – Users can provide a common description no matter what language the plugin is implemented in.

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

This feature provides the general extensibility mechanism needed by the PQE external search agent, PQE filter and the object qualification's user definable operator feature.

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

This problem is not currently being addressed. So far, each feature that has needed extendibility has invented its own mechanism.

Which languages must support this capability?

C++, Java, C# and Python.

Which platforms must be supported?

All.

Do any competitors already have this capability?

Yes; Oracle has generalized its extendibility as Cartridges.

Customers who require this capability:

Those wishing to extend our product, such as those wishing to extend queries to external data sources, add user defined indexes, add encryption, etc.

Revenue at risk or which could be won:

Many new customers are coming to us because we can solve their difficult problems, but to do that best they will often want to extend our product.

When is this required?

R10, as features depending on this feature are scheduled for that release.

Review

Feature Sizing

Efforts	*Size*
Development	
QA	
Documentation	

Scheduled for Objy Release

Assigned Engineering Group