HRAC RFP Submission: Resolved Issues

Title	**Access Control**
ID	1
Description	1. What is the model/mechanism? 2. Is the model/mechanism fixed or extensible? If extensible, how so? 3. Does the rules of the model/mechanism use resource content as security metadata?
Date Issued	11/18/98 Date Resolved 10/ 7/98
Depends on Issues	8, 9
Pointed by	John Barkley
Related Refs	
To propose a resolution	
Resolution description	The initial submission text provides the object model, interfaces and description of the mechanisms

---

Title	**Resource Security Metadata**
ID	8
Description	I can see the following 3 ways to obtain resource security metadata (I use words "metadata" and "data" to mean the same type of data unless specified otherwise): 1. Pass only resource id to the ADO. In order to obtain the data the ADO is supposed to go elsewhere and use resource id to find the data. 2. Pass only resource id to the ADO and use it as a carrier of the data. Where as, a. data syntax and semantics of the data are predefined and assumed. b. data syntax is not assumed. Data is represented by parsable tag-like structures. Semantics of data is predefined elsewhere. c. syntax and semantics of data are defined elsewhere and a reference to those definitions is passed along the data itself. Each way has pros and cons. What one (or more than one) should be used in this submission?
Date Issued	8/10/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	[HRAC resources] thread in the mail list of the submitting team
To propose a resolution	
Resolution description	Approach #1 was chosen by the initial submission

Title	**Resource Identifier Structure**
ID	9
Description	What syntax and semantics should the resource identifier have?
Date Issued	8/10/98 Date Resolved 10/ 7/98
Depends on Issues	8
Pointed by	Carol Burt
Related Refs	[HRAC resources] thread in the submission team mail list + minutes from July 30 mee
To propose a resolution	
Resolution description	The submission does not have notion of resource identifiers. It manipulates only with resource names. A resource name is a sequence of strings. It does not have predefined semantics

Title	**Understanding of application functionality or data**
ID	10
Description	Should HRAC understand application data/functionality?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	14
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	
Resolution description	HRAC should have no understanding of application functionality or data.

Title	**Definition of "Resource"**
ID	11
Description	What is a resource?
Date Issued	8/11/98 Date Resolved
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	
Resolution description	Define "resource" exactly how it is defined in the HRAC RFP: "a 'secured resource' can be any valuable asset of an application owner, which is accessed by an application on behalf of a principal using it, and access to which is to be controlled according to the owner's interests."

Title	**Definition of "Resource Name"**
ID	12
Description	What is a resource name?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	11
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	Bob Burt
Resolution description	It is an identifier for an application defined resource. Its format is as follows type sequence<string> ResourceName;

Title	**Definition of "Resource Metadata"**
ID	13
Description	What is resource metadata?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	11
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	Bob Burt
Resolution description	Resource metadata is data that describes a resource. Note: "describes" does not mean provides the value of the resource. HRAC has no requirement to maintain, obtain, or use this meta data. If it exists or is used, it is strictly an application issue. Resource metadata is accessed/interpreted in the scope of policy evaluators, dynamic attribute evaluators, and it is out of scope of the HRAC service.

Title	**Information passed to the decision maker logic**
ID	14
Description	What information does an application pass to the decision maker logic?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	Bob Burt
Resolution description	An application service (ADO client) passes: 1. A resource name 2. An operation name 3. Principal security attributes

Title	**Operation Format**
ID	15
Description	What is the format of an operation?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	14
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	Bob Burt
Resolution description	typedef string Operation;

Title	**Authorization rule specification**
ID	16
Description	How are rules specified? Bob Burt: I propose that each ResourceName-Operation pair has the following associated with it: 1. One or more credential attributes 2. A sequence of time-pairs Permission is granted if "any" of the user credential attributes match "any" of the associated credential attributes and the current time does not fall in one of the time-pairs period of time. If resources are used in a hierarchical fashion, that is, the resource name has more than one name in it's sequence, then sub-setting of rules should be enforced. One might want to further define something called a policy object that would be a matrix of operations x attributes. The policy object could have a name an be used in the future without rebuilding the matrix.
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	
Resolution description	The submission does not specify authorization rules. Instead, it provides mechanisms that allow ADO to invoke different policy evaluators that can implement various authorization models.

Title	**Goals for Initial submission**
ID	17
Description	What else is needed for an initial submission?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	10-16
Pointed by	Bob Burt
Related Refs	msg00108.html
To propose a resolution	
Resolution description	see the initial submission text

Title	**Contents of a resource reference**
ID	19
Description	Should the contents of a resource reference be opaque or implementation-dependant?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	14, 20
Pointed by	John Barkley
Related Refs	minutes of the conference call of August 11, 1998
To propose a resolution	
Resolution description	No resource reference is in the submission model

Title	**Definition of "Resource Reference" term**
ID	20
Description	What does term "resource reference" mean?
Date Issued	8/11/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	August 11, 1998, conference call minutes
To propose a resolution	Bob Blakley
Resolution description	The submitted response does not use a concept "resource reference" at all.

Title	**Word "operation" is reserved in CORBA**
ID	23
Description	This is the AccessDecision interface. I'm requesting that we not use the term "operation" as an identifier for the first parameter. This is because it can be confused with an operation on a CORBA interface. This is clearly one way that the parameter might be used, but this type of operation might also be considered a subclass resource and the operation would be "use". And it has little meaning when the resource is actually a parameter of a CORBA operation.
Date Issued	9/11/98 Date Resolved 1/10/99
Depends on Issues	No dependencies
Pointed by	Carol Burt
Related Refs	http://cadse.cs.fiu.edu/omg/hrac-rfp/pdf00004.pdf
To propose a resolution	Carol Burt
Resolution description	I got over it. Carol

Title	**Legacy access control engines**
ID	24
Description	There are examples of legacy access control engines which would not fit well with the interfaces proposed by Bob Blakley
Date Issued	8/25/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Bret Hartmen and Juggy
Related Refs	msg00118.html
To propose a resolution	Bret to provide examples
Resolution description	The submission provides mechanisms to replace or add additional access control engines

Title	**Deny/grant semantics**
ID	29
Description	Where there is a resource, world is denied, group is granted, specific individual of group is denied. Cannot be handled by current interfaces.
Date Issued	9/15/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	The new (as of 10/07/98) HRAC design model does not define a particular access policy mechanism. Instead, it provided a way to have multiple access policy evaluators.

Title	**Time-dependant rights**
ID	30
Description	Do we need to capture periodic rights? For example, after hours security may be more strict
Date Issued	9/15/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	The new (as of 10/07/98) HRAC design model does not define a particular access policy mechanism. Instead, it provided a way to have multiple access policy evaluators.

Title	**Possible holes in negative states of access policies**
ID	31
Description	There are possible holes in the current access policy model as of when the issue is raised.
Date Issued	9/15/08 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	The new (as of 10/07/98) HRAC design model does not define a particular access policy mechanism. Instead, it provided a way to have multiple access policy evaluators.

Title	**Resource key and POA-based call-backs**
ID	32
Description	In application services using POA, an object key can be used by the servant manager to incarnate an appropriate servant or even to have one servant for all call-back object. In this case we do not need resource key when calling dynamic attribute service
Date Issued	9/15/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Konstantin Beznosov
Resolution description	The issue was raised because of the assumption that the resource key would be used only when ADO calls back the application service that invoked the ADO. The assumption was not correct.

Title	**Removing resource subtrees and nodes**
ID	33
Description	Should we allow two methods, one to remove a resource node and one to remove a resource name subtree?
Date Issued	9/15/98 Date Resolved
Depends on Issues	No dependencies
Pointed by	Bart de Greef
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	administering resources is out of scope of this spec

Title	**Time associated with dynamic attributes**
ID	35
Description	Should the dynamic attribute method support the notion of time, so that a relationship can be remembered" to exist in the past and not only at the present time?
Date Issued	9/15/98 Date Resolved
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	Decided to not address in spec

Title	**Locality constrainness of ADO**
ID	5
Description	Should an Access Decision Object to be locality constrained?
Date Issued	8/10/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	
To propose a resolution	
Resolution description	No, it should not. Only security attributes are passed through methods of the ADO interface. So, it's safe to have ADO ¬ locality constrained.

Title	**Quality of Protection as an authorization decision factor**
ID	6
Description	Should current quality of protection policy information in ADO client be used as a factor in authorization decisions as principal credentials are?
Date Issued	8/10/98 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	msg00055.html -- msg00057.html
To propose a resolution	
Resolution description	No explicit support in the proposed specification.

Title	**Required rights expression**
ID	22
Description	Taking into account that the authorization rule language in HRAC is expected, at list by me, to be more rich in its expressiveness than the one provided by CORBA security access model, I contend that just simple expression of required rights (such as "A or B or C or D" or "A and B and C and D") would not suffice. I believe that more complex expressions should be allowed.
Date Issued	9/11/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	Any type of policy evaluator can be used within HRAC service. The required-rights model is not used by the mandatory part of the submission any more.

Title	**Type of the policy name**
ID	27
Description	Should policy name be opaque, string, or something else. Policy name is something which depends on a particular policy evaluator. For some policy evaluators, policy name being a string is not convenient at all.
Date Issued	10/ 7/98 Date Resolved
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	

Title	**Typedefs in the interface modules**
ID	28
Description	When interfaces in IDL compiled into Java classes and interfaces, "typedef" statements generate dummy classes that are very inefficient to use comparatively to original types. Is there a way to avoid/solve this problem in the submitted interfaces of the spec?
Date Issued	9/15/98 Date Resolved 10/ 7/98
Depends on Issues	No dependencies
Pointed by	Bart de Greef
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	The submission team decided that this problem belongs to the scope of IDL-> Java mapping and if it is a serious issue for developers/designers, it should be addressed through IDL->Java mapping RTF or via a new RFP in the OMG. The submission team decided to use typedef mechanisms as appropriate.

Title	**Multiple use of resource name**
ID	34
Description	Add text to the submission explaining that real world resource can be defined and used multiple times. There is no way to prevent it
Date Issued	9/15/98 Date Resolved 1/10/99
Depends on Issues	No dependencies
Pointed by	
Related Refs	Minutes of the submitters meeting in Seattle on September 15, 1998
To propose a resolution	Nobody Assigned
Resolution description	administration of resource name space is out of scope of this spec

Title	**All policy evaluators have to be consulted**
ID	36
Description	Since the policy combinator is handed only results of the policy evaluation, all PolicyEvaluators have to be consulted before a final decision can be made. The work of DecisionCombinator cannot be optimized. For example, even in the case when the very first decision from a PolicyEvaluator makes the "final call" all other PolicyEvaluators have to be consulted.
Date Issued	11/10/98 Date Resolved 2/18/99
Depends on Issues	No dependencies
Pointed by	Polar Humenn
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	See austin-99 minutes

Title	**Grouping of resources for associating them with policies**
ID	37
Description	How can resources of a particular types to be associated with particular policy or policies?
Date Issued	11/10/98 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Ed from NIST
Related Refs	No additional references
To propose a resolution	Konstantin
Resolution description	with ResourceNamePatterns. See Austing-99 minutes.

Title	**Inference of sensetive information via accessing particular non-sensetive resources**
ID	38
Description	What if a client would access 3 different resources associtated with information of low sensetivity level and then use the obtained information to infere information of higher level of sensetivity?
Date Issued	11/10/98 Date Resolved 11/10/98
Depends on Issues	No dependencies
Pointed by	Jarom Soeller
Related Refs	No additional references
To propose a resolution	Bob Blakley
Resolution description	The RFP did not ask to address this kind of issue

Title	**Removing safely PolicyEvaluator**
ID	39
Description	How can a client of an implementation of the PolicyEvaluatorLocatorAdmin interface safely remove an evaluator? The interface does not have a delete_evaluator(s) operation, so I would assume that the recommended technique would be to: 1. Get sequence of evaluators via PolicyEvaluatorLocator::get_policy_decision_evaluators. 2. Build a new PolicyEvaluatorList removing the unwanted evaluator. 3. Invoke PolicyEvaluatorLocatorAdmin::replace_evaluators with the new list. The problem is what happens when another client invokes add_evaluators or replace_evaluators in the time frame between steps 1 and 3 above. I would suggest that other client will be quite unhappy.
Date Issued	1/21/99 Date Resolved 2/18/99
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	Added delete_evaluators(). See Austin-99 minutes.

Title	**safely restoring a list of default evaluators**
ID	40
Description	How can a client of an implementation of the PolicyEvaluatorLocatorAdmin interface safely restore a list of default evaluators? Since the PolicyEvaluatorLocatorAdmin::set_default_evaluators does not return a list of the current default evaluators, it will be impossible to reliably return the list of default evaluators to their previous state. (Trying to get the default list by invoking get_policy_decision_evaluators with a bogus resource name seems to have the same "non-locking problem" as described in Issue 39
Date Issued	1/21/99 Date Resolved 2/18/99
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	issue #39
To propose a resolution	Nobody Assigned
Resolution description	set_default_evaluators() returns now old evaluators list. See Austin-99 minutes and new IDL code.

Title	**safely removing a DecisionCombinator for a given resource name**
ID	41
Description	How can a client of an implementation of the PolicyEvaluatorLocatorAdmin interface safely remove a DecisionCombinator for a given resource name? There appears to be no way to do this.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	Added delete_combinator(). See Austin-99 minutes.

Title	**safely restoring the default DecisionCombinator**
ID	42
Description	How can a client of an implementation of the PolicyEvaluatorLocatorAdmin interface safely restore the default DecisionCombinator? This issue is similar to Issue 40.
Date Issued	1/21/99 Date Resolved 2/19/09
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	See #40 resolution.

Title	**invalid object reference for a PolicyEvluator and/or DecisionCombinator**
ID	43
Description	What is an implementation of the AccessDecision interface supposed to do when the PolicyEvaluatorLocator returns an invalid/unreachable object reference for a PolicyEvluator and/or DecisionCombinator? The specification makes no demands on implementations of the PolicyEvaluator and DecisonCombinator interfaces. That is, it does not dictate whether they must be "transient" or "persistent", whether they must maintain state or not, and whether they can be launched by an ORB or must be manually launched. For example, based on the specification, it is perfectly legal to build an implementation of the PolicyEvaluator interface that is “transient” and must be manually launched. This means the IOR for this implantation is only valid for the life of the process that created it and the process can only be manually launched. If the process is killed and its clients (e.g. ADO’s) continue to try to use it, they will get CORBA System Exceptions (COMM_FAILURE in OrbixWeb). The client has no way of determining that the object reference is valid or invalid, the process is running or not running, or whether the network path is available or not available. The real issue here seems to be that the client (e.g. ADO) has no way of notifying the PolicyEvaluatorLocator that a problem exists and that perhaps it should investigate the fact that it is distributing a potentially bogus object reference.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	ADO raises an exception to the ADO client. See Austin-99 minutes. We do not mandate persistence. It should be an implementation feature.

Title	**state persistently in PolicyEvaluatorLocatorAdmin and PolicyEvaluatorAdmin**
ID	44
Description	Clients of the PolicyEvaluatorLocatorAdmin and PolicyEvaluatorAdmin interfaces cannot determine whether or not the implementations of those interface maintain state persistently. This is important to a client because it must know whether of not it needs to reset the state to that which it expects. Without this knowledge, the client code would need to be changed to migrate from a persistent to a non-persistent implementation. If the client code must change between implementations, why have a specification?
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	We do not mandate persistency. We assume that implementation will have some persistence characterisitcs. E.g. storing policy evaluator location information somewhere. We left details to implementors.

Title	**scalability of the proposed spec implementations**
ID	45
Description	Can complete implementations of this specification actually be made to be "scalable"? From the experience that I have had, users of security control systems expect minimal if not transparent overhead from such a system. This is often not considered a problem during the design phases, but when implementations begin to adversely affect response times or overall use of system resources ($$$), the problem becomes real. When one looks at the Access Decision Model (2.3.1), it appears that the Access Decision will make a minimum of four operation invocations. This in itself could result in unacceptably excessive overhead. Even more importantly, one should consider what would comprise an implementation of a PolicyEvaluatorLocator interface. This interface must maintain a collection of “resource name – evaluator” associations. Each time that it returns evaluators, it must query this collection to determine if the requested resource name matches one or more of those “resource name – evaluator” associations. It must query this collection for every request for a list of evaluators. If this collection is small enough to be held in its entirety in memory, this can be a somewhat compute-intensive process. If, however, the system has scaled to the extent that this collection can no longer be effectively maintained in memory, one must now endure to extremely high overhead of doing external storage based queries. Perhaps one could invent some form of caching technique that might optimize this query; however, it appears to me that this might not be possible. It appears that the "entire" collection must be queried to ensure that there are no "resource name -- evaluator" associations. Implementation of these external storage based queries would seem to create overhead that would be unacceptable to all but the least demanding users.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Bob Burt
Related Refs	No additional references
To propose a resolution	Nobody Assigned
Resolution description	By moving evaluators behind decision combinator, we elminated at least one call. Plus we introduced resource name grouping via patterns, which should enable scalable and efficient implementations and management. Sane implementors would co-locate all HRAC objects inside of process or capsule.

Title	**Policy Evaluators are not interchangeable**
ID	48
Description	Issue 3. Since Policy is not defined, the idea that Policy Evaluators are interchangeable seems disingenuous. It seems that if policy interface is non-standard, then policy evaluators are non-standard, by definition. Thus, the proposed standard for interoperation doesn't succeed in providing interoperability.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Kurt Schurenberg
Related Refs	No additional references
To propose a resolution	Bob Blakley
Resolution description	We did not mean to make them interchangable. "Broken as designed."

Title	**no extra capabilities than what CORBA security has**
ID	49
Description	Issue 4. Little or no new capabilities appear to be provided. I understand that this specification is designed to work with or without an existing CORBA security layer. However, it would appear to me that what the specification does is rework the CORBA security capabilities to produce the same capabilities in a different way. That different way includes a loss of object orientation due to use of named resources, which seems like a negative to me. There doesn't appear to be an ability to do any more than was already available via the standard CORBA security. Since you aren't using OO for policies or the objects controlled, but just name strings, it's possible that things can be defined at a different level here, but I'm not sure that's a positive change.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Kurt Schurenberg
Related Refs	No additional references
To propose a resolution	Bob Blakley
Resolution description	All the capabilities, except one, required by the RFP are provided. The spec does provide capabilities that CORBASEC does not. Using resource names instead of OR was a mandatory requirement.

Title	**Decision Combinator overtakes things which ought to be defined in policy**
ID	50
Description	Issue 5. The Decision Combinator overtakes things which ought to be defined in policy. It seems to me that a security policy defines how to combine multiple incoming elements of an authorization decision. When you add the capabilities to modify the decision combinator to make more complex grouping and evaluation of individual policy evaluation results, you will have started to define the policy interface that you are trying to avoid defining. Again, though, you are starting to require that the policy be embedded in code, rather than metadata, which makes the system extremely expensive to maintain.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Kurt Schurenberg
Related Refs	No additional references
To propose a resolution	Bob Blakley
Resolution description	DC is a policy (just different) object.

Title	**Specified system requires high overhead**
ID	51
Description	Issue 6. System appears to require a high overhead. By ignoring the admittedly difficult issue of defining a policy interface, the submission has been forced to create a complex set of objects which may be evaluating policy on details down to the attribute level. If a search has produced a list of objects, each of which contain 3 attributes which are policy-controlled, then you must go through the 3 object interfaces four times for every element on the list. This will be too slow to be commercially acceptable.
Date Issued	1/21/99 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Kurt Schurenberg
Related Refs	No additional references
To propose a resolution	Bob Blakley
Resolution description	PC answer: attribute granularity policies are not a side-affect. They are intended. This is what we were explicitely asked to do. The name space is up to you.

Title	**ADO interfaces Exceptions**
ID	2
Description	What exceptions should be raised by ADO's methods? Should it be the matter of a policy whether ADO raises an exception when something goes wrong or silently denies access to a resource? Three possible directions are identified: 1. Methods raise no exceptions 2. Methods raise exceptions a. Methods raise only system exceptions (like NO_PERMISSION, BAD_PARAM, NOT_IMPLEMENT) b. Methods raise system and application exceptions,
Date Issued	8/11/98 Date Resolved 2/19/99
Depends on Issues	
Pointed by	Konstantin Beznosov
Related Refs	mail list archive messages # msg00040.html, msg00054.html
To propose a resolution	
Resolution description	See the minutes of Austin-99 meeting

Title	**Exception(s) raised by multiple_action_access_allowed() method in ADO interface**
ID	4
Description	From her message: "Should access decision methods throw exceptions at all... an audit log should have this info... but not the client... seems it should be a binary decision." Derived from a conference call discussion: How would a programmer use an exception returned by multiple_action_access_allowed() method? Is not it better return any problem indications in the returned sequence instead of raising an exception?
Date Issued	8/10/98 Date Resolved 2/19/99
Depends on Issues	2
Pointed by	Carol Burt
Related Refs	
To propose a resolution	
Resolution description	ADO client should do access_allowed() on each request individually if it recieves an exception while invoking multiple_access_allowed().

Title	**Consistent Terminology**
ID	7
Description	Can we define some consistent Terminology? Define Evaluator and Policy Name
Date Issued	8/10/98 Date Resolved 2/19/99
Depends on Issues	11, 12, 13
Pointed by	Carol Burt / John B.
Related Refs	msg00039.html
To propose a resolution	David Chizmadia to fill in secti
Resolution description	We are happy now with the terminology we have.

Title	**Correct name of the specified functionality: no "access control" but "authorization decisions"**
ID	18
Description	"HRAC" stands for healthcare resource access control. Clearly, the functionality for which the RFP is asking (and what a submission is supposed to specify) is concern only with making authorization decisions, i.e. no actual access control is in the scope of the RFP. Thus, the specified functionality should be renamed from "access control" to something else that would reflect the fact that it specifies only authorization decision part, i.e. not control.
Date Issued	8/11/98 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	msg00110.html
To propose a resolution	
Resolution description	Changed to Resource Access Decision

Title	**Facility or Service**
ID	21
Description	Should the final functionality be called a "facility", "service", or something else?
Date Issued	8/11/98 Date Resolved 2/19/99
Depends on Issues	No dependencies
Pointed by	Konstantin Beznosov
Related Refs	No additional references
To propose a resolution	
Resolution description	It is a facility but it is not reflected directly in the name.