Integration Specification

Resources API v7.1

Prepared By: IRWIN Core Team

Last Updated: 12/20/2021

Contents

1 Introduction

The IRWIN Resources API is a separate set of methods for external systems to use in addition to the Incidents API. The Resource data integration is focused on sharing resource data related to incident dispatch, ordering and statusing. The resource API is structured to facilitate data integration between external system in 3 main areas:

1. Managing resources (overhead, aircraft, equipment, teams, crews and modules).
2. Sharing resource requests and statusing for initial attack and extended attack.
3. Recording overhead resource experience for currency maintenance by qualification systems.

Business areas supported include (but is not limited to) Dispatch, Property Management, Personnel Qualification and Training, Resource Ordering, and other incident management related areas. Resource integration in this context includes specifically Overhead, Equipment, Aircraft, Crews, Teams and Modules.

The Resources API is used in conjunction with the Incidents API (*Integration Specification – Incidents API v7*).

1.1 Purpose and Audience

The Resource Integration Specification introduces and expands upon those topics necessary to begin data exchange through the IRWIN RESOURCES API. A formal discovery process is required to obtain an authentication credential, which allows access to both the IRWIN Incidents and Resources API. This document is not a replacement for that process.

The IRWIN Community is comprised of the IRWIN Core Team and IRWIN Extended Teams. The IRWIN Core Team is responsible for developing and supporting the technical integration based on requirements provided by the wildland fire community. The IRWIN Core team is comprised of technical developers, data architects, business leads and implementation leads. IRWIN Extended Teams represent the technical and business persons who support a system that exchanges data with other systems through the IRWIN Integration Service.

This document is intended for extended teams and particularly their system developers responsible for modifying their application for data exchange within the IRWIN Resource integration services.

1.2 Points of Contact

Chuck Wamack – IRWIN DOI Business Lead

dwamack@ios.doi.gov

208.334.6190

Kara Stringer – IRWIN Forest Service Business Lead

kstringer@usda.gov

801.531.5320

Brandon Green - IRWIN Project Manager

Brandon_Green@ios.doi.gov

410.303.3307

2 Conceptual Architecture

The IRWIN Resources API is designed to broker common operational RESOURCE data across various wildland fire applications. This RESTful API exposes standard **Add, Query, Update, Delete** and utility operations, allowing integrated systems to share operational data. Although the API is customized, it follows standard extension guidelines of the underlying ArcGIS for Server software. These custom operations:

- Validate data standards
- Enforce updates by authoritative systems only on a role basis
- Provide operations specific to the business needs of the wildland fire community

The API's role is to provide the ability for many disparate systems to create and edit resource information, or retrieve updated data on demand. With the understanding that these systems leverage different core technologies, languages, platforms, are in varying lifecycle stages, or have different business rules, the API provides a common, flexible approach to integration yet NWCG accepted standards and business workflows.

3 Development Considerations

This section serves as a guidance to developers in understanding the main areas of coding for interacting with the Resources API to exchange data. Figure 1 below depicts both the Incident Feature Service layer and the Resource Feature Service layer and their related tables. Incident layer/tables are depicted in GREEN and the Resources layer/tables in BLUE – including their relationships to each other. Domain tables are depicted in YELLOW. Focused views of this diagram are provided in the following sections.

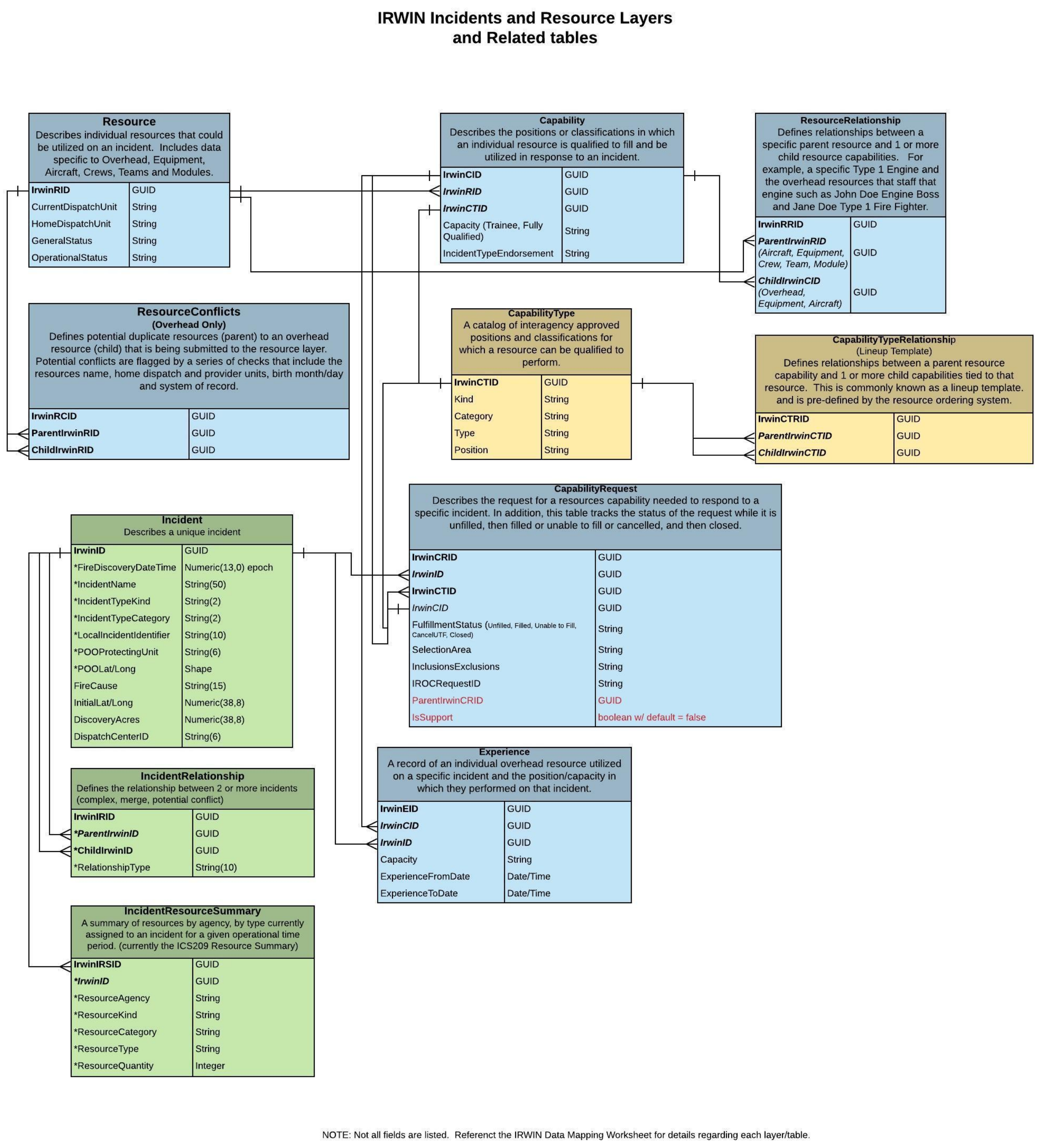

IRWIN Incidents and Resource Layers and Related tables
Resource
Describes individual resources that could be utilized on an incident. Includes data specific to Overhead, Equipment, Aircraft, Crews, Teams and Modules.
IrwinRID GUID
CurrentDispatchUnit String
HomeDispatchUnit String
GeneralStatus String
OperationalStatus String
Capability
Describes the positions or classifications in which an individual resource is qualified to fill and be utilized in response to an incident.
IrwinCID GUID
IrwinRID GUID
IrwinCTID GUID
Capacity (Trainee, Fully Qualified) String
IncidentTypeEndorsement String
ResourceRelationship
Defines relationships between a specific parent resource and 1 or more child resource capabilities. For example, a specific Type 1 Engine and the overhead resources that staff that engine such as John Doe Engine Boss and Jane Doe Type 1 Fire Fighter.
IrwinRRID GUID
ParentIrwinRID (Aircraft, Equipment, Crew, Team, Module) GUID
ChildIrwinCID (Overhead, Equipment, Aircraft) GUID
ResourceConflicts
(Overhead Only)
Defines potential duplicate resources (parent) to an overhead resource (child) that is being submitted to the resource layer. Potential conflicts are flagged by a series of checks that include the resources name, home dispatch and provider units, birth month/day and system of record.
IrwinRCID GUID
ParentIrwinRID GUID
ChildIrwinRID GUID
CapabilityType
A catalog of interagency approved positions and classifications for which a resource can be qualified to perform.
IrwinCTID GUID
Kind String
Category String
Type String
Position String
CapabilityTypeRelationship
(Lineup Template)
Defines relationships between a parent resource capability and 1 or more child capabilities tied to that resource. This is commonly known as a lineup template. and is pre-defined by the resource ordering system.
IrwinCTRID GUID
ParentIrwinCTID GUID
ChildIrwinCTID GUID
CapabilityRequest
Describes the request for a resources capability needed to respond to a specific incident. In addition, this table tracks the status of the request while it is unfilled, then filled or unable to fill or cancelled, and then closed.
IrwinCRID GUID
IrwinID GUID
IrwinCTID GUID
IrwinCID GUID
FulfillmentStatus (Unfilled, Filled, Unable to Fill, CancelUTF, Closed) String
SelectionArea String
InclusionsExclusions String
IROCRequestID String
ParentIrwinCRID GUID
IsSupport boolean w/ default = false
Incident
Describes a unique incident
IrwinID GUID
*FireDiscoveryDateTime Numeric(13,0) epoch
*IncidentName String(50)
*IncidentTypeKind String(2)
*IncidentTypeCategory String(2)
*LocalIncidentIdentifier String(10)
*POOProtectingUnit String(6)
*POOLat/Long Shape
FireCause String(15)
InitialLat/Long Numeric(38,8)
DiscoveryAcres Numeric(38,8)
DispatchCenterID String(6)
IncidentRelationship
Defines the relationship between 2 or more incidents (complex, merge, potential conflict)
IrwinIRID GUID
*ParentIrwinID GUID
*ChildIrwinID GUID
*RelationshipType String(10)
Experience
A record of an individual overhead resource utilized on a specific incident and the position/capacity in which they performed on that incident.
IrwinEID GUID
IrwinCID GUID
IrwinID GUID
Capacity String
ExperienceFromDate Date/Time
ExperienceToDate Date/Time
IncidentResourceSummary
A summary of resources by agency, by type currently assigned to an incident for a given operational time period. (currently the ICS209 Resource Summary)
IrwinIRSID GUID
*IrwinID GUID
*ResourceAgency String
*ResourceKind String
*ResourceCategory String
*ResourceType String
*ResourceQuantity Integer
NOTE: Not all fields are listed. Referenct the IRWIN Data Mapping Worksheet for details regarding each layer/table.

Figure 1 Incident and Resource Layers and Related Tables

- Incident
  - Describes an individual incident such as Unique Incident Identifier, Fire Discovery Date & Time, Incident Name, Point of Origin Latitude/Longitude and Fire Cause. There are over 125 Incident data elements, with only 14 of which are required by a CAD to create

an incident.

- Incident Relationships
  - Defines relationships between two or more incidents such as Complexes, Merges and Potential Duplicates.

- Incident Resource Summary
  - A summary of resources assigned to an incident for a given operational time period.

- Resource
  - Describes individual resources that could be utilized on an incident. Includes data specific to Overhead, Equipment, Aircraft, Crews, Teams and Modules.

- Resource Conflicts
  - Defines potential conflicts of an overhead resource that is being submitted (add or update feature) to the resource layer. Potential conflicts are flagged by a series of checks that include the resources name, home dispatch and provider units, birth month/day and system of record.

- Capability Type
  - A catalog of interagency approved positions and classifications for which a resource can be qualified to perform.

- Capability
  - Describes the positions or classifications in which an individual resource is qualified to fill and be utilized in response to an incident.

- Capability Type Relationships
  - Defines relationships between a parent resource capability and 1 or more child capabilities tied to that resource. This is commonly known as a lineup template. For example, a Type 1 Engine and the overhead resources that staff that engine such as the Engine Boss and the Type 1 Fire Fighters.

- Resource Relationships
  - Defines relationships between a specific parent resource and 1 or more child resource capabilities tied to that resource. For example, a specific Type 1 Engine and the overhead resources that staff that engine such as John Doe Engine Boss and Jane Doe Type 1 Fire Fighter.

- Capability Request
  - Each record describes the request for a resources capability needed to respond to a specific incident. In addition, this table tracks the status of the request while it is unfilled, then filled or unable to fill or cancelled, and then closed.

- Experience
  - A record of an individual overhead resource utilized on a specific incident and the position/capacity in which they performed on that incident.

3.1 Authentication and Authorization

All integrated systems are provided a system level account to authenticate with the IRWIN API. Systems will authenticate by acquiring a short-lived token string via the `GenerateToken` operation and adding its value to a `token` parameter when making any succeeding web calls to IRWIN. As part of the `GenerateToken` response, the token's expiration time is provided. Once the token's expiration time is met, the integrated system needs to request a new token.

> NOTE: The maximum token lifetime that may be requested in IRWIN is 60 minutes. A token should not be requested more than twice per hour. Best practice is requesting once every hour.

When generating a token, a parameter named 'client' is supplied. There are several options for this parameter as described in the online documentation. It is recommended that systems use the 'referer' client as the supplied parameter. This option avoids issues such as IPs that may change between requests of getting the token and subsequent calls that use the token, and is a more stable option in environments where IPs may not always remain the same.

To implement this, when the token is requested, the client value is supplied as 'referer', and a value is supplied for the optional 'referer' parameter. This value for 'referer' can be any value, but will need to be supplied on subsequent calls that use the token and the two values must match.

Sample input:
username='yoursystem',
password='yourpassword',
client='referer',
referer='YOUR REFERER VALUE',
expiration=60,
f=json

On subsequent calls that use this token, the token value must be supplied, and the REFERER header value in any HTTP requests must match the value for 'referer' provided in the token request.

Documentation for the GenerateToken operation can be found at:
http://resources.arcgis.com/en/help/arcgis-rest-api/index.html#/Generate_Token/02r3000000ts000000/

Once a credentialed system connects to IRWIN, access to individual API operations and data elements is based on authorization roles. Each integrated system is placed into a role defined during the discovery process. To access the **Resource** methods, a system must be granted one or more of the following roles in addition to the appropriate incidents role(s).

For a full list of available roles specific to resources, reference the table below.

IRWIN Resource Authorization Roles

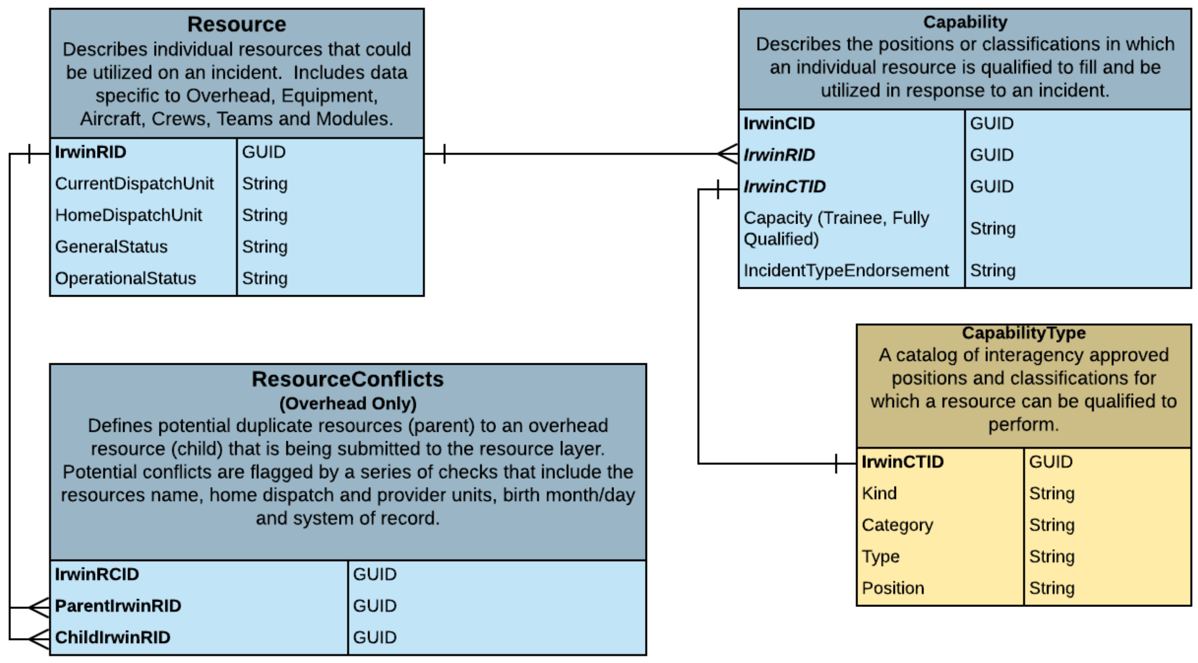
Resource API Role	Detail
Ordering	Role for systems that perform resource ordering functions. Grants access to create and update non-overhead resources and their capabilities and also create and update resource requests.
Qualification	Role for systems that manage overhead resource qualifications. Grants access to create and update overhead resources and read resource experience.
Lineup	Role for systems that manage a resource lineups and role call for initial attack purposes. Grants access to create and update resource relationships.
Dispatch	Role for systems that dispatch resources to perform initial attack. Grants access to create and update resource capabilities, capability requests and resource status.
Capability Type Admin	Role for maintenance of the capability type domain within IRWIN. Grants access to create, update and delete capability types.
Resource Read	Role for reading resources and request data. Grants read access to the resource layer and related tales.

3.2 Resources and Capabilities

3.2.1 Resource Layer and Related Tables

The resource feature layer and related tables are utilized for maintenance of a repository of resources and their capabilities for deployment on a wildland fire. In order to add and update a resource and their capabilities, it is important to understand the relationship between the resource feature layer and the related tables pertinent to performing these actions. The figure below depicts the keys that relate each of the tables. The CapabilityType table is a domain for capabilities. Use this table to establish the IrwinCTID when adding capabilities.

Resource Feature Layer and Related Tables


Resource
Describes individual resources that could be utilized on an incident. Includes data specific to Overhead, Equipment, Aircraft, Crews, Teams and Modules.
IrwinRID GUID
CurrentDispatchUnit String
HomeDispatchUnit String
GeneralStatus String
OperationalStatus String
Capability
Describes the positions or classifications in which an individual resource is qualified to fill and be utilized in response to an incident.
IrwinCID GUID
IrwinRID GUID
IrwinCTID GUID
Capacity (Trainee, Fully Qualified) String
IncidentTypeEndorsement String
CapabilityType
A catalog of interagency approved positions and classifications for which a resource can be qualified to perform.
IrwinCTID GUID
Kind String
Category String
Type String
Position String
ResourceConflicts
(Overhead Only)
Defines potential duplicate resources (parent) to an overhead resource (child) that is being submitted to the resource layer. Potential conflicts are flagged by a series of checks that include the resources name, home dispatch and provider units, birth month/day and system of record.
IrwinRCID GUID
ParentIrwinRID GUID
ChildIrwinRID GUID

NOTE: Not all fields are listed. Referenct the IRWIN Data Mapping Worksheet for details regarding each layer/table.

3.2.2 Key Data Concepts for Resources and Capabilities

For both reading and updating resources and their capabilities, there are a few data elements that are fundamental to understanding the IRWIN resource data. Please reference the IRWIN Data Mapping Workbook for more detailed information on these data elements.

3.2.1 Resource

- Shape - Geographic representation of resource's physical location. This location is submitted as a geometry object. If available to the external system, these coordinates can represent the position from a resources tracking device. If a tracking device is not available, then the coordinates could be derived from an organizational address associated with the resource. When a resource's OperationalStatus is updated to 'At Incident', if no geometry is provided, IRWIN will update the resource's SHAPE to the SHAPE of the incident for that resource's 'Filled' CapabilityRequest.

- LocationTimeStamp - this field can be used in conjunction with the shape to indicate the date/time of the resource's last known location. This field is not required however.

- ResourceKind - The resource layer contains all operational resources that can respond to a fire. This field indicates the kind of resource (Equipment, Crew, Aircraft, Overhead, Team, Module, Equipment Group, Aircraft Group, Overhead Group). In the data mapping workbook, you can find which data elements apply to and are required for adding and updating each resource kind.

- IsValid - Indicates whether the resource is valid within IRWIN. Valid resources are records from the Provider and Dispatch Center having primary responsibility for the resource. Invalid resources are a result of duplicates or invalid entries.
  - When IsValid = 0 on a resource, IRWIN will delete any ResourceConflict records and also set IsValid = 0 for any capability for that resource.
  - When reading or updating resource records, filtering for IsValid = 0 should be taken into account as appropriate for the intended results. These records will also be marked for deletion by IRWIN.

- ResourceSOR - The irwin userid of the system that currently is the System of Record for the resource and their capabilities (i.e. IQCS). The value "None" is also valid and is used for transferring individuals to another System of Record. Once the ResourceSOR is set to a system userid, it can only be updated to 'None' and likewise only be updated from 'None' to a system userid.

- IsQuarantined - Indicates whether a resource of ResourceKind "Overhead" is potentially conflicting with one or more overhead resources based on IRWIN conflict detection rules. When submitting resources, use this field to detect if the incident has been flagged as a potential duplicate. If it has, your system will need to present conflict resolution options to a user. Quarantined resources will not be visible to any other integrated systems. See Resource Conflicts section for more details.

- GeneralStatus - Indicates whether the resource is available or unavailable to be ordered. The dispatch centers responsible for the resource can set the general status. Also, once a resource is ordered, the general status may be set to "Unavailable" by IRWIN based on business rules related to the Operational Status.

- OperationalStatus - Describes the status of the resource as they are utilized by an incident. The dispatch centers through their CAD's or IROC can set the operational status of resources once they are on a filled capability request. Reference the ***Resource Requests*** section for more details on the dependencies between creating/updating Capability Requests and setting the operational status for a resource.

- OperationalName - The CURRENT common operational reference to a Resource, the name of the resource and/or call sign. The IROC application will provide and update the OperationalName for non-overhead resources (equipment, crews, modules and teams). IROC may update the OperationalName based on how the resource is currently being utilized through a filled request.

- Capability table - In order for a resource to be utilized on an incident, they must have at least one capability assigned to them.

- Text fields do NOT allow characters ">" or "<", as those values could cause problems when ingested by other systems. For example ">>" can be interpreted as HTML by other systems, resulting in an error.

3.2.2 Capability

- IsValid - Indicates if the capability is a valid record for the resource. If false, indicates that the capability is not and never was valid for the resource.

- IsActive - Indicates if the capability is currently active for the resource. If false, indicates that the capability was active at one time, but is no longer active for the resource.

- Capacity - The capacity in which the overhead resource is qualified to perform a position - either as a trainee or fully qualified. This data element can be updated.

- NeededByDateTime - Date/time expressed in epoch format that the resource is needed by. Required on CapabilityRequest Add.

3.2.3 Reading Resources and Capabilities

Reading resource and capability data is accomplished through the ArcGIS REST Query feature service layer operation referencing the resource feature layer number (0). This generic read operation allows for a wide variety of spatial and SQL where clause queries to be executed against the underlying data, returning an array of matched features. Query will accept IrwinRIDs, ResourceKind, OperationalStatus or any other search criteria in the form of a where clause, as well as specify which fields to return.

Documentation for Query can be found at:

http://resources.arcgis.com/en/help/arcgis-rest-api/02r3/02r3000000r1000000.htm

3.2.3.1 Additional Parameters

As part of the SOI (Server Object Interceptor), the API includes enhancements to the COTS (Commercial off-the-shelf) **Query** operation making custom request parameters available:

- Resource Layer
  - includeCapabilities=true (default is false)
    - includeCapabilityType=true (default is true if includeCapabilities=true)
  - includeExperiences=true (default is false)
  - includeRelationships=true (default is false)
  - includeConflicts=true (default is false)
- Capability Layer
  - includeResource=true (default is false)
- Capability Request Layer

- includeResource=true (default is false)
- includeCapability=true (default is false)
    - includeCapabilityType=true (default is true if includeCapability=true)

Including any of the above parameters will enhance the response results by including the corresponding objects as properties of each result. For example, appending "includeCapabilities=true" to a Resource query will cause each resource object in the response to have an additional resource object-level property, "capabilities", which is an array of Resource Capability objects associated with that resource.

3.2.4 Resource/Capability Creation

Resources and Capabilities can be created using the ArcGIS REST AddFeatures operation. This generic create operation allows for individual or batch features to be created against the underlying data layer. AddFeatures will accept one or more standard feature objects, which express the Resource(s) the user wishes to create.   Depending on the kind of resource being submitted, it is required to express a minimum number of data elements to successfully create the resource. Reference the *IRWIN Data Mapping Workbook* for details regarding required fields, data types, valid values and validation rules.

All submitted data elements are run against validation in order to enforce data standards. A successful creation will result in a response indicating success and the resource payload (i.e., IrwinRId, and values of auto-calculated data elements) for the integrated system to act on. If the AddFeatures operation fails validation, an error response object is returned.

In addition, IRWIN will determine if the resource being submitted is potentially in conflict with a resource that already exists. For non-overhead resources, a conflict will result in a failed request. For overhead resources, potential conflicting resources relationships will be written to the ResourceConflicts table and the resource being submitted will be quarantined.

Documentation for AddFeatures can be found at:

http://resources.arcgis.com/en/help/arcgis-rest-api/02r3/02r30000010m000000.htm

3.2.5 Resource Updates

Resources and Capabilities are updated in IRWIN using the ArcGIS REST UpdateFeatures operation. This generic update operation allows for individual or batch features to be updated against the underlying data layer. UpdateFeatures will accept one or more standard feature objects, which express the Resource(s) the user wishes to update.

Documentation for UpdateFeatures can be found at:

http://resources.arcgis.com/en/help/arcgis-rest-api/02r3/02r3000000zt000000.htm

3.2.5.1 Additional Parameters

As part of the SOI, the API includes enhancements to the COTS UpdateFeatures operation making custom request parameters available. For the resource being updated (such as an engine), then additional resource values can be automatically updated for children as defined in the ResourceRelationship table.

- Resource Layer
  - applyToChildren=field1,field2,...fieldN

Including applyToChildren=field1,field2,...fieldN will cause the SOI to update the existing resource records for each child and apply the same value(s) for the field(s) specified.

Note: *To use this custom parameter:*

1. *the ResourceRelationships MUST exist between the initial resource being updated and its children.*
2. *the Resource records MUST exist for the children in the relationship.*

3.3 Overhead Resource Transfer/Remove Scenarios

Resources often move from one organization to another. In addition, resources also need to be removed or no longer work for an agency. The table below describes resource transfer and remove scenarios and the data changes that need to be made for each.

NOTE: *A person should not be transferred if they are currently on assignment. The SOR should not be set to 'NONE' if the OperationalStatus is NOT "Returned from Assignment" or NULL.*

Transfer/Remove Scenarios	ResourceSOR/Capability	Organization Fields	GeneralStatus
Scenario 1: REMOVE (release) - Resource is no longer valid and has been set to isValid = false.	System of Record sets: ResourceSOR = 'NONE' System of Record set Capability.isActive = false and Capability.isValid = false for all of the resource's capabilities.	No change.	System of Record sets: GeneralStatus = 'Unavailable'
Scenario 2: Resource transfers to another organization that is still managed by the ResourceSOR.	No change	Organization fields updated to unit where resource is transferred: HomeDispatchUnit HomeUnit ProviderUnit ManagerContactInfo [and any other pertinent data like jet port, email, etc.]	No change or updated as necessary.

Scenario 3: Resource is released and could be transferred to another organization that is managed by a different System of Record (or could be managed again by the same SOR)	Old System of Record set ResourceSOR to 'NONE' and set all quals to Capability.isActive = false New System of Record sets ResourceSOR from 'NONE' to their Irwin system of record. New System of Record sets Cabability.isActive = true for all valid capabilities (also add new capabilities if applicable)	Organization fields updated to unit where resource is transferred: HomeDispatchUnit HomeUnit ProviderUnit ManagerContactInfo [and any other pertinent data like jet port, email, etc.]	New System of Record updates GeneralStatus = 'Available'

3.4 Resource Conflicts

3.4.1 Non-Overhead Resources

For entry of non-overhead resources the following rules will ensure uniqueness. If there is an exact match found that already exists, the record will not be successfully submitted. See Section 4 for error codes.

- Aircraft: the TailNumber must be unique.
- Equipment: If the VIN is NOT Null the VIN must be unique AND If the SerialNumber is (NOT NULL) the SerialNumber must be unique.
- Crews, Teams, Modules: The OperationalName plus the HomeDispatchUnit must be unique.

3.4.2 Overhead Resources

For adding or updating overhead resources, the following rules will ensure uniqueness. If there is an exact match found that already exists, the record will not be successfully submitted (error response). For potential duplicates that need further inspection, the record will be quarantined. When a resource is "quarantined", IRWIN sets the IsQuarantined = 1 (true) and creates a record in the ResourceConflicts table for each existing record matching a quarantine rule below.

- NameFirst plus NameLast plus NameMiddle plus HomeDispatch plus ProviderUnit plus BirthMonthDay must be unique.
- If NameFirst plus NameLast plus NameMiddle plus BirthMonthDay match, the resource is quarantined.
- If all but one of the following are the same, the resource is quarantined - NameFirst plus NameLast plus NameMiddle plus HomeDispatch plus ProviderUnit plus BirthMonthDay.

- If the BirthMonthDay plus the NameLast match, the resource is quarantined.
- If the BirthMonthDay plus the NameFirst plus NameMiddle (first letter) match, the resource is quarantined.

3.4.3 Resolving Overhead Conflicts

The CreatedBySystem should present the quarantined resource alongside the parent(s) for resolution. This is accomplished by querying the ResourceConflicts table for the ChildIrwinRID of the resource in quarantine. Pertinent information to display to help the user determine the appropriate outcome should include:

- BirthMonthDay
- HomeDispatchUnit, HomeUnit, ProviderUnit, ProviderAgency
- ManagerContactInfo
- NameFirst, NameLast, NameMiddle
- ResourceClearinghouseID (if available)
- ResourceSOR

There are two scenarios for resolving conflict:

Scenario 1 - Child Lost (is not a valid resource, resource is the same person as the potential duplicate)

The CreatedBySystem should:

1) Set the child resource to isValid = False (0)
2) When a quarantined resource is updated from IsValid = 1 (true) to IsValid = 0 (false), IRWIN will automatically delete any ResourceConflict records where the updated Resource is the child in the ResourceConflict.

   The child's resulting Resource and ResourceConflicts record should look like:

Resource IsQuarantined	Resource IsValid	ResourceConflicts
1 (True)	0 (False)	Irwin deletes conflict record related to parent and child.

Scenario 2 - Both the Parent and the Child Win (both are valid distinct persons)

In this scenario, the child resource is still a different and valid person from the parent resource. The CreatedBySystem should:

1) Set the child resource to IsQuarantined = 0 (False).
   a) NOTE: *Do not send any other fields in this update request, only include IsQuarantined. Including other fields will result in either an error or IsQuarantined not being updated. This is a known issue.*

2) When a resource is updated from IsQuarantined = 1 (true) to IsQuarantined = 0 (false), IRWIN will automatically delete any ResourceConflict records where the updated Resource is the child in the ResourceConflict.

   Note: IsQuarantined cannot be nulled.

   The child's resulting Incident record should look like:

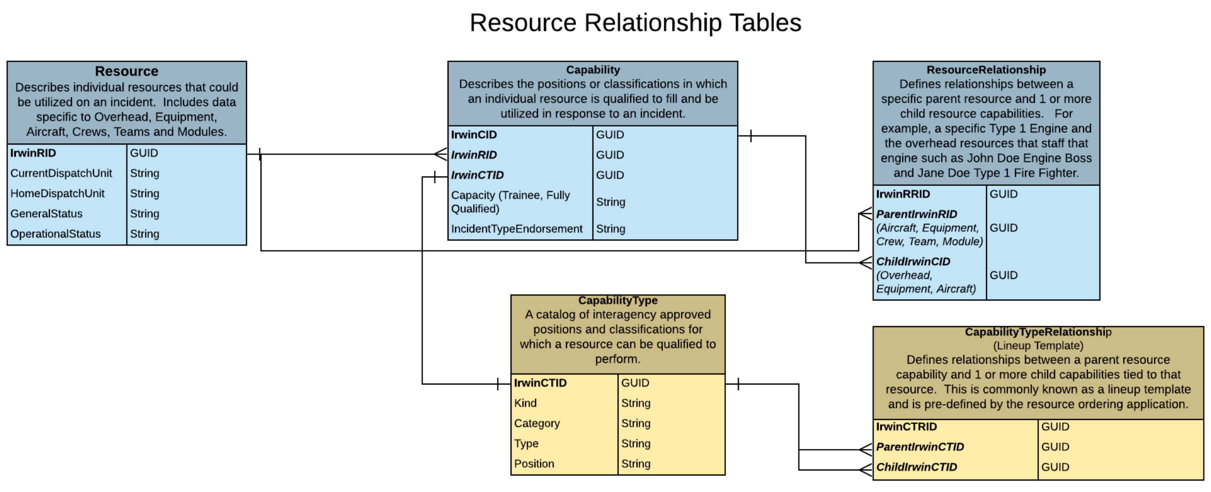
Resource IsQuarantined	Resource IsValid	ResourceConflicts
0 (False)	1 (True)	Irwin deletes relationship for parent/child combination.

3.5 Resource Relationships

Resources can be related to one another for purposes of creating daily line-ups and rostering. In order to establish relationships between a resource(s), it is important to understand the relationship between the resource feature layer and the related tables pertinent to forming relationships. The figure below depicts the keys that relate each of the tables. There are 2 types of relationships that can be leveraged:

- CapabilityTypeRelationship - use this table to populate a lineup with the predefined configurations of capabilities which are provided by IROC. This table is read-only.

- ResourceRelationships - use this table to establish a connection between a specific resource (from the resources table) to one or more specific resource capabilities. Once this relationship is established, updates to the parent can be cascaded to the children using the custom "applyToChildren" parameter defined in the *Resource Updates* section.


Resource Relationship Tables
Resource
Describes individual resources that could be utilized on an incident. Includes data specific to Overhead, Equipment, Aircraft, Crews, Teams and Modules.
IrwinRID GUID
CurrentDispatchUnit String
HomeDispatchUnit String
GeneralStatus String
OperationalStatus String
Capability
Describes the positions or classifications in which an individual resource is qualified to fill and be utilized in response to an incident.
IrwinCID GUID
IrwinRID GUID
IrwinCTID GUID
Capacity (Trainee, Fully Qualified) String
IncidentTypeEndorsement String
ResourceRelationship
Defines relationships between a specific parent resource and 1 or more child resource capabilities. For example, a specific Type 1 Engine and the overhead resources that staff that engine such as John Doe Engine Boss and Jane Doe Type 1 Fire Fighter.
IrwinRRID GUID
ParentIrwinRID (Aircraft, Equipment, Crew, Team, Module) GUID
ChildIrwinCID (Overhead, Equipment, Aircraft) GUID
CapabilityType
A catalog of interagency approved positions and classifications for which a resource can be qualified to perform.
IrwinCTID GUID
Kind String
Category String
Type String
Position String
CapabilityTypeRelationship
(Lineup Template)
Defines relationships between a parent resource capability and 1 or more child capabilities tied to that resource. This is commonly known as a lineup template and is pre-defined by the resource ordering application.
IrwinCTRID GUID
ParentIrwinCTID GUID
ChildIrwinCTID GUID

NOTE: Not all fields are listed. Referenct the IRWIN Data Mapping Worksheet for details regarding each layer/table.

3.5.1 Deleting Resource Relationships

Resource relationships are deleted using the ArcGIS REST **DeleteFeatures** operation. The ResourceRelationship table does NOT have an isValid data element that most other Irwin tables use for effectively deleting records.

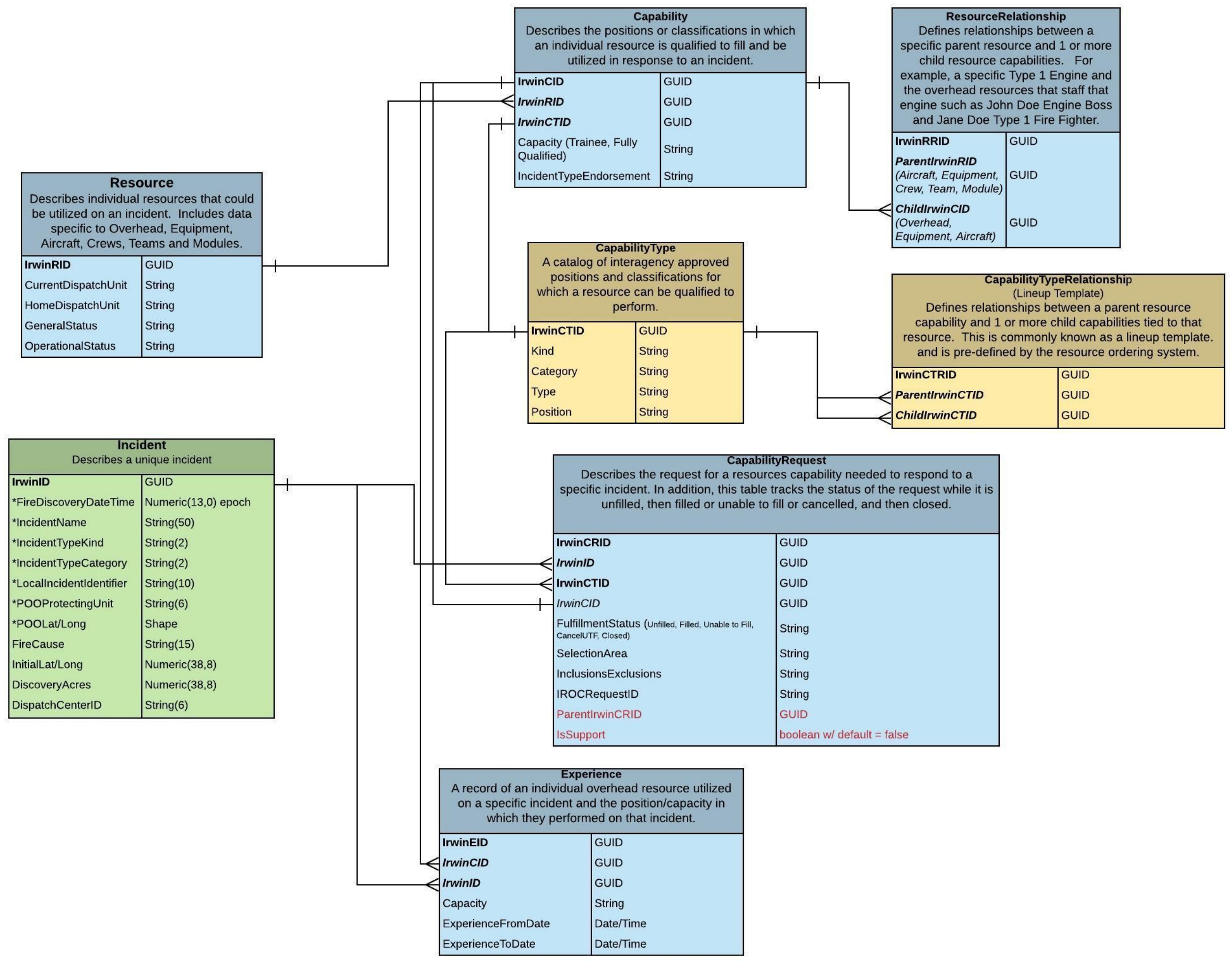
3.6 Resource Requests

The resource request structure focuses on sharing the actual status of a resource and their capability once ordered for an incident and also in what capacity that resource is serving. The main business workflows supported by the IRWIN resource request services for sharing data include:

- Providing initial attack resource capabilities for an incident.
- Requesting/ordering additional resource capabilities to respond to an incident.
- Filling resource capability requests.
- Tracking the operational status of a resource from mobilization to an incident to de-mobilization .

The diagram below shows how the capability request table is tied to the incident and resource feature layers and the capability table.

IRWIN Resource Requests and Related Tables



Capability
Describes the positions or classifications in which an individual resource is qualified to fill and be utilized in response to an incident.

IrwinCID	GUID
IrwinRID	GUID
IrwinCTID	GUID
Capacity (Trainee, Fully Qualified)	String
IncidentTypeEndorsement	String

ResourceRelationship
Defines relationships between a specific parent resource and 1 or more child resource capabilities. For example, a specific Type 1 Engine and the overhead resources that staff that engine such as John Doe Engine Boss and Jane Doe Type 1 Fire Fighter.

IrwinRRID	GUID
ParentIrwinRID *(Aircraft, Equipment, Crew, Team, Module)*	GUID
ChildIrwinCID *(Overhead, Equipment, Aircraft)*	GUID

Resource
Describes individual resources that could be utilized on an incident. Includes data specific to Overhead, Equipment, Aircraft, Crews, Teams and Modules.

IrwinRID	GUID
CurrentDispatchUnit	String
HomeDispatchUnit	String
GeneralStatus	String
OperationalStatus	String

CapabilityType
A catalog of interagency approved positions and classifications for which a resource can be qualified to perform.

IrwinCTID	GUID
Kind	String
Category	String
Type	String
Position	String

CapabilityTypeRelationship
(Lineup Template)
Defines relationships between a parent resource capability and 1 or more child capabilities tied to that resource. This is commonly known as a lineup template. and is pre-defined by the resource ordering system.

IrwinCTRID	GUID
ParentIrwinCTID	GUID
ChildIrwinCTID	GUID

Incident
Describes a unique incident

IrwinID	GUID
*FireDiscoveryDateTime	Numeric(13,0) epoch
*IncidentName	String(50)
*IncidentTypeKind	String(2)
*IncidentTypeCategory	String(2)
*LocalIncidentIdentifier	String(10)
*POOProtectingUnit	String(6)
*POOLat/Long	Shape
FireCause	String(15)
InitialLat/Long	Numeric(38,8)
DiscoveryAcres	Numeric(38,8)
DispatchCenterID	String(6)

CapabilityRequest
Describes the request for a resources capability needed to respond to a specific incident. In addition, this table tracks the status of the request while it is unfilled, then filled or unable to fill or cancelled, and then closed.

IrwinCRID	GUID
IrwinID	GUID
IrwinCTID	GUID
IrwinCID	GUID
FulfillmentStatus (Unfilled, Filled, Unable to Fill, CancelUTF, Closed)	String
SelectionArea	String
InclusionsExclusions	String
IROCRequestID	String
ParentIrwinCRID	GUID
IsSupport	boolean w/ default = false

Experience
A record of an individual overhead resource utilized on a specific incident and the position/capacity in which they performed on that incident.

IrwinEID	GUID
IrwinCID	GUID
IrwinID	GUID
Capacity	String
ExperienceFromDate	Date/Time
ExperienceToDate	Date/Time

NOTE: Not all fields are listed. Reference the IRWIN Data Mapping Worksheet for details regarding each layer/table.

3.6.1 Reading Resource Requests

Reading resource request data is accomplished through the ArcGIS REST Query feature service layer operation referencing the Capability_Request feature layer number (2). This generic read operation allows for a wide variety of spatial and SQL where clause queries to be executed against the underlying data, returning an array of matched features. Query will accept IrwinIDs, FulfillmentStatus, IrwinCID's or any other search criteria in the form of a where clause, as well as specify which fields to return. Depending on the type of information that a system needs returned, the related tables in the diagram above may need to be joined to the query results by the indicated keys or use the custom parameters as defined below.

- ★ When querying CapabilityRequest include IsValid =1 (true) in the where clause.

As part of the SOI (Server Object Interceptor), the API includes enhancements to the COTS (Commercial off-the-shelf) Query operation making custom request parameters available:

- Resource Layer
    - includeCapabilities=true (default is false)
        - includeCapabilityType=true (default is true if includeCapabilities=true)
    - includeExperiences=true (default is false)
    - includeRelationships=true (default is false)
    - includeConflicts=true (default is false)
- Capability Layer
    - includeResource=true (default is false)
- Capability Request Layer
    - includeResource=true (default is false)
    - includeCapability=true (default is false)
        - includeCapabilityType=true (default is true if includeCapability=true)

3.6.1.1 Query Subordinate requests for requested resource

When requests are created, IROC (the resource ordering system) assigns request id's (IrocRequestID) to each capability request record that reflects the parent resource (ie E-1, A-1) and then subordinate requests to that resource (ie E-1.1, E-1.2, etc). Subordinate requests are given a ParentIrwinCRID value that identifies the parent request for that subordinate. This value is either populated by IRWIN when initial attack requests are submitted from a CAD using the "apply to children" parameter or the ParentIRWINCRID is populated by IROC for rostered requests. To query for a request and all subordinate requests, use the following logic:

- Query CapabilityRequest - order by IrwinID, IrocRequestID, ParentIrwinCRID, . Within each set of IrwinID's you will get all the orders for that incident. Then you can determine the parent request as those without a "dot" number and the ParentIrwinCRID is null. The subordinates for each parent would be all records that contain ParentIrwinCRID of the parent request.

3.6.2 Creating and Updating Resource Requests

CapabilityRequests can be created using the ArcGIS REST AddFeatures operation.

CapabilityRequests can be updated using the ArcGIS REST UpdateFeatures operation.

CapabilityRequests may be submitted against quarantined incidents, but will not be shared with all integrated systems until conflict is resolved.

CapabilityRequests <u>should not</u> be sent for quarantined incidents older than 24 hours. The CAD systems will implement this rule. This rule will not be enforced in IRWIN.

The order in which adds and updates are sent when creating and filling capability requests is important to ensure the updates are successful and that the resource's experience is properly documented. The following guidelines should be followed:

1. The initial CapabilityRequest record requires:
    a. capability type id (IrwinCTID)
    b. IrwinID for the incident
    c. FulfillmentStatus
    d. SelectionArea, and InclusionsExclusions
    e. RequestedCapacity is also required if FulfillmentStatus is not "Filled"

2. If the FulfillmentStatus = "Filled", the add or update must also include a capability id (CID) to tie the request to a resource and their qualification. For initial attack resources, the CAD should also update the mobilization ETD and ETA times when the request is filled.

3. A resource can only be on ONE "Filled" request for non-preposition type incidents. A resource (overhead or non-overhead) can be on more than one "Filled" capability request if both are NOT preposition incidents.This allows for a resource to have a filled request on a preposition incident and then also be requested on a non-preposition incident.

4. <u>Set a resource's OperationalStatus to "At Incident" AFTER creating the filled request for that resource otherwise an error will result.</u>

5. To close out a request (Mob then Demob scenario)
    a. There are two mechanisms in use for this functionality today and presented below
        i. Mechanism #1
            1. CAD sets the resource's OperationalStatus to one of the following
                a. Demob En Route, Reassigned (At Incident), Released (At Incident), Returned From Assignment.
                b. Set the "applyToChildren=OperationalStatus, Shape" if appropriate
            2. CAD updates the CapabilityRequest setting:
                a. DemobETA = current + N (value N may vary by CAD)
                b. DemobETD to current date/time
                c. ETA to current date/time
            3. IROC sets internal request status to Released
            4. IROC will not generate Experience in this scenario
            5. CAD updates Resource GeneralStatus to Available and OperationalStatus to Null
            6. CAD updates CapabilityRequest setting DemobETA to current date/time and FulfillmentStatus to Closed
        ii. Mechanism #2
            1. CAD updates the CapabilityRequest setting:
                a. DemobETA = current + N (value N may vary by CAD)
                b. DemobETD to current date/time
                c. ETA to current date/time

2. IROC will use the DemobETA value to determine the OperationalStatus as either "Demob En Route" or "Return from Assignment"
3. IROC will set the GeneralStatus to "Available"
4. IROC updates CapabilityRequest FulfillmentStatus = "Closed"
5. CAD updates Resource OperationalStatus = "Null"

6. To Reassign a Resource within current Dispatch
    a. CAD creates a new unfilled CapabilityRequest (B)
        i. CAD sets FulfillmentComment field = IrwinCID of the reassigned resource
    b. IROC updates the original CapabilityRequest (A) setting FulfillmentStatus to Reassign/Closed (Reassign in IROC maps to Closed in IRWIN)
    c. IROC updates the new CapabilityRequest (B) setting IrwinCID = value from FulfillmentComment and FulfillmentStatus = Filled
        i. IROC creates subordinate requests as applicable
    d. CAD updates CapabilityRequest (B) with ETA and ETD values

7. To Cancel a request that was entered in error
    a. Set Resource FulfillmentStatus = "Canceled"
    b. Set isValid = 0 (false)

3.6.2.1 Additional Parameters

As part of the SOI, the API includes enhancements to the COTS AddFeatures operation making custom request parameters available. If the request being created is filled with a resource (such as an engine) and that engine has been rostered, additional requests can be automatically created for the children.

- Resource Capability Request Layer
    - applyToChildren=FulfillmentStatus,IsValid,SelectionArea,InclusionsExclusions,IsContractResourceAllowed

Including applyToChildren (for FILLED requests only) will cause the SOI to create a new capability request record for each child and apply the same value(s) for the field(s) specified. IRWIN will also add the ParentIrwinCRID to the record for each child capability request created.

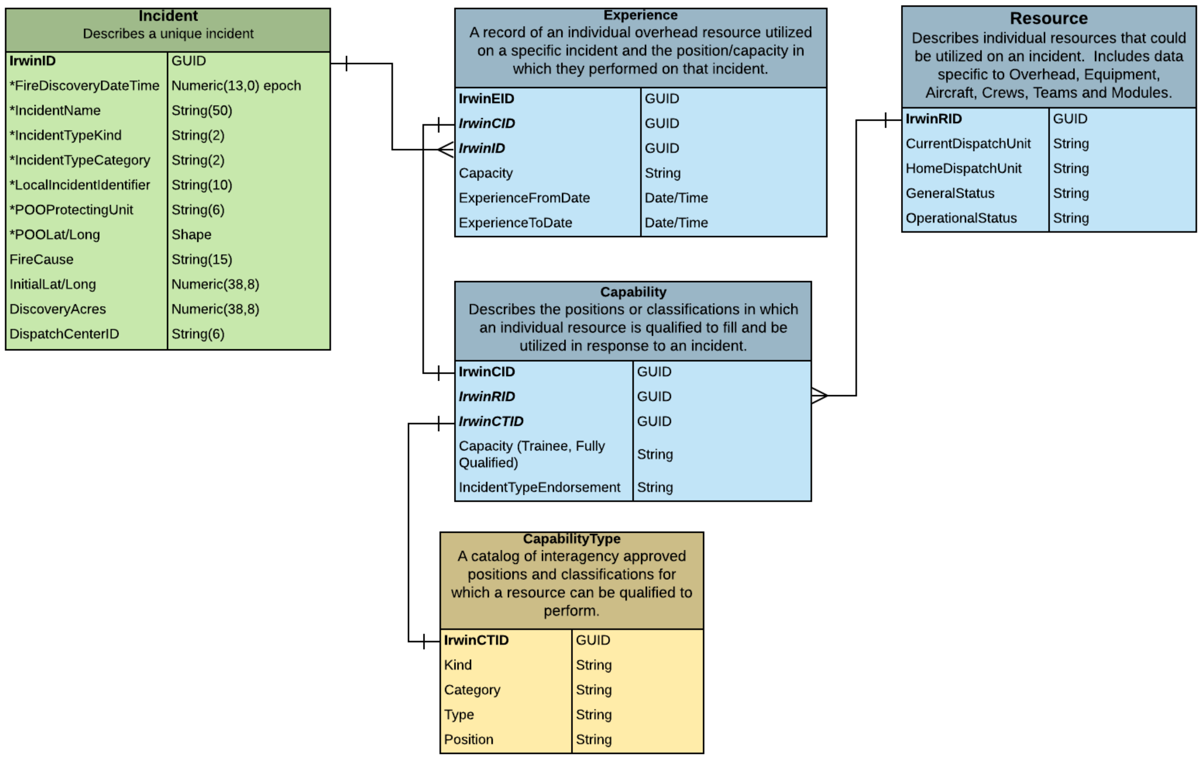
Note: *To use this custom parameter, the ResourceRelationships MUST exist between the initial resource being requested and its children. In addition, the request must be created with a status of "FILLED".*

3.7 Resource Experience

Resource experience is created and updated by IROC as a by-product of the resource request workflows. Qualification systems can read the experience records to maintain currency for the resources to which they are the system of record. The diagram below shows the keys that join the experience for a resource to an incident for querying purposes.

★ Qualification systems should query for experience records where isValid = 1 (true) AND ExperienceToDate is not null.

Experience and RelatedTables



Incident Describes a unique incident	
IrwinID	GUID
*FireDiscoveryDateTime	Numeric(13,0) epoch
*IncidentName	String(50)
*IncidentTypeKind	String(2)
*IncidentTypeCategory	String(2)
*LocalIncidentIdentifier	String(10)
*POOProtectingUnit	String(6)
*POOLat/Long	Shape
FireCause	String(15)
InitialLat/Long	Numeric(38,8)
DiscoveryAcres	Numeric(38,8)
DispatchCenterID	String(6)

Experience A record of an individual overhead resource utilized on a specific incident and the position/capacity in which they performed on that incident.	
IrwinEID	GUID
IrwinCID	GUID
IrwinID	GUID
Capacity	String
ExperienceFromDate	Date/Time
ExperienceToDate	Date/Time

Resource Describes individual resources that could be utilized on an incident. Includes data specific to Overhead, Equipment, Aircraft, Crews, Teams and Modules.	
IrwinRID	GUID
CurrentDispatchUnit	String
HomeDispatchUnit	String
GeneralStatus	String
OperationalStatus	String

Capability Describes the positions or classifications in which an individual resource is qualified to fill and be utilized in response to an incident.	
IrwinCID	GUID
IrwinRID	GUID
IrwinCTID	GUID
Capacity (Trainee, Fully Qualified)	String
IncidentTypeEndorsement	String

CapabilityType A catalog of interagency approved positions and classifications for which a resource can be qualified to perform.	
IrwinCTID	GUID
Kind	String
Category	String
Type	String
Position	String

NOTE: Not all fields are listed. Referenct the IRWIN Data Mapping Worksheet for details regarding each layer/table.

3.7.1 Resource Experience Creation Method (as communicated to IRWIN by IROC)

IROC creates and updates Experience records for a Resource based on filled resource requests once the Resource has departed the Incident. Experience creation is triggered when both IROC's (internal) MobEndDate and DemobStartDate have non-null values on a capability request and DemobStartDate is after MobEndDate. IROC will set the ExperienceToDate = DemobStartDate and ExperienceFromDate = MobEndDate. If IROC's MobEndDate is further into the future or equals DemobStartDate, meaning the resource was not actually at the incident, IROC will not generate resource experience.

4 Error Handling

The IRWIN API returns a variety of indicators and status codes detailing the success or failure of actions. Upon addFeatures or updateFeatures, a boolean "success" property is returned, indicating if the action was successful or not. If false, an error property is also returned which lists the error **code** (indicating the kind of error) and **description** (providing the actual error messages).

Additional documentation for error responses can be found at:
http://resources.arcgis.com/en/help/arcgis-rest-api/02r3/02r300000017000000.htm

4.1 Validation Errors

If the request results in one or more validation errors, the response will include an "error" object with the "code" property specified as 8004. The "description" property of the error object will be an array of validation error objects. Each validation error that is relevant will be included as a separate object with one of the following codes and messages.

JSON Syntax:

```
{
  "objectId":  <objectId>, //int, objectId value of the updated/inserted feature
  "globalId":  <globalId>, //string, string globalId value of updated/inserted feature
  "success":  <true | false>, //boolean, false if edit was not applied
  "error": { //only returned if success is false
    "code":  <code>, //integer, error code
    "description": [ //array of validation error objects
      {
        "error": {
          "code":  <code>, //integer, error code
          "message":  <message>, //string, validation error message/description
          "conflictObjectId": <conflictObjectId> //integer, only returned if validation
error is a "unique" conflict
        }
      }
    ]
  }
}
```

In the following tables, text highlighted in gray represents example values only; the actual text may vary based on the input and/or context.

Code	Example message	What does it mean?	How to fix it
101	value (This is wrong!) must be composed of alphanumeric, hyphen, or period characters.	The value may only contain letters, numbers, hyphens (-), and/or periods (.)	Remove any characters from the value that are not letters, numerical digits, hyphens, or periods.
103	value (X) must be an accepted value ([A|B|C]).	The value must be one of a defined list (or "domain").	Ensure the value you are passing matches one of the specified values in the domain values for this field. Note that case may be important.
105	Invalid type. Expected number.	The value must be a number; spaces, letters, or other non-numerical characters are not allowed. Periods and hyphens may be allowed if they are contextually relevant, such as for floating point or negative values.	Ensure the value is a number with no spaces.
	value (10) must not exceed 5.	The numerical value must be less than or equal to the stated maximum.	Lower the value to less than or equal to the maximum.
106	value (abcdefg) must not exceed 6 characters.	The value is too long.	Shorten the length of the value.

107	Invalid type. Expected number.	The value must be a number; spaces, letters, or other non-numerical characters are not allowed. Periods and hyphens may be allowed if they are contextually relevant, such as for floating point or negative values.	Ensure the value is a number with no spaces.
	value (1) must exceed 5.	The numerical value must be more than or equal to the stated minimum.	Raise the value to more than or equal to the minimum.
108	value (tooshort) must be at least 15 characters.	The value is too short.	Lengthen the value to be at least the minimum length; spaces are not valid padding. Typically this means left-padding the value with zeroes.
109	value is not nullable.	The value cannot be omitted on addFeatures or nullified on updateFeatures.	For addFeatures requests, you must include a non-null value. For updateFeatures, ensure the value is not being set to "null".
110	value contains disallowed characters.	The value contains characters that are not allowed for this field, such as spaces or special characters.	Check the value to ensure it contains only characters relevant to this field data.
111	a value is required for IrwinCAD systems.	(Systems with IrwinCAD role only) This value is required on addFeatures.	Ensure the value is not missing or null.
112	a value is required.	This value is required.	Ensure the value is not missing or null.

113	value (XXWRNG) must begin with a valid state code.	The first two characters of this string value must be a valid state abbreviation (or, in certain cases, "CA" or "MX").	Ensure the first two characters are a valid NWCG-standard state code (or "CA" or "MX", if the relevant data falls within Canada or Mexico accordingly)
114	value (123) must be type string.	The value must be a string, and cannot be passed as a JSON-specified type of boolean, number, array, or object.	Ensure the value is enclosed by quotes.
	value (3.14) must be type integer.	The value must be a whole number.	Ensure the numerical value is a whole number with no decimal.
	value (wrong) must be type float.	The value must be a number.	Ensure the value is a number.
	value (Sunday, 23 Feb 2019) must be type epoch datetime (long integer).	The value must be a valid datetime value in Unix-time (or "epoch"-time) format. See this link.	Ensure that the value is a datetime value, expressed as a whole number of milliseconds after 12:00 am, January 1, 1970. This will be a 13-digit number.
	value ([34.01,-117.34]) must be type geometry.	The value was not recognized as a correctly formatted JSON geometry. See this link.	Ensure the value is a correctly formed JSON object, with (at a minimum) a value for "x" and "y".
	value ({"lat":34.01,"lon":-117.34}) must be a correctly formed geometry object. See https://developers.arcgis.com/documentation/common-data-types/geometry-objects.htm#POINT	The value was not recognized as a correctly formatted JSON geometry. See this link.	Ensure the value is a correctly formed JSON object, with (at a minimum) a value for "x" and "y".

900	IrwinID is invalid.	The IRWIN API couldn’t find or parse the specified ID value. That is, the ID was not found in the request.	Ensure the IrwinID/ IrwinRID/ IrwinRSID is a valid GUID, specified as a string value with no leading/trailing braces.
901-904	IrwinID (069BA152-C519-4502-A560-3F72754FB862) not found.	The IRWIN API was unable to find a record in the relevant table/layer with the specified ID.	Ensure the IrwinID/ IrwinRID/ IrwinRSID is a valid GUID and refers to an existing record in the relevant layer.
905	Error querying for IrwinID 069BA152-C519-4502-A560-3F72754FB862: Database error	Depending on the error, this may indicate a server failure of some kind.	Check your input data and try again. If this happens repeatedly, please report it to the IRWIN implementation team.

Code	Example message	What does it mean?	How to fix it
300	value (069BA152-C519-4502-A560-3F72754FB862) must be an existing IrwinCID in ResourceCapability.	The record in a related layer couldn't be resolved from the GUID provided.	Ensure the ID is a valid GUID, specified as a string value with no leading/trailing braces, for the relevant related layer.
	Error querying for IrwinCID in ResourceCapability (069BA152-C519-4502-A560-3F72754FB862): Database error	Depending on the error, this may indicate a server failure of some kind.	Check your input data and try again. Depending on the error, this may indicate a server failure of some kind. If this happens repeatedly, please report it to the implementation team.
301	value (069BA152-C519-4502-A560-3F72754FB862) must be an existing IrwinID in Incident.	The value of IrwinID must refer to an existing incident in the Incident layer of the Irwin service.	Ensure the IrwinID is a valid GUID, specified as a string value with no leading/trailing braces, for the Incident layer of the Irwin service.

	value (069BA152-C519-4502-A560-3F72754FB862) cannot be used because the referenced object has IsValid=0.	The ID provided is a valid GUID and refers to an existing object in the related layer, but the target record that it refers to has IsValid=0 and thus cannot be used for the attempted request.	Ensure the ID is a valid GUID, specified as a string value with no leading/trailing braces, for the relevant related layer.
	Error querying for IrwinID in Incident (069BA152-C519-4502-A560-3F72754FB862): Database error	Depending on the error, this may indicate a server failure of some kind.	Check your input data and try again. Depending on the error, this may indicate a server failure of some kind. If this happens repeatedly, please report it to the implementation team.
302	If value is not null, it can only be set to null.	If value is not null, it can only be set to null.	Set the value to null or remove the field from the request.
303	If value is not 'NONE', it can only be set to 'NONE'.	If value is *not* set to the value identified in the message, it can only be set to that value.	Set the value to the specified value or remove the field from the request.
304	Unable to validate whether value is unique because the value of 'IrwinRID' could not be parsed.	Another field that the validator depends on couldn't be read correctly (in this case, IrwinRID).	Ensure all required fields are present in the request and conform to the respective documented requirements.
	value of 1 must be unique for (IrwinRID) value(s) in ResourceCapability (conflicts with IrwinRID '069BA152-C519-4502-A560-3F72754FB862').	Although a given reference ID (in this case, IrwinRID) may exist multiple times in the given related layer (in this case, ResourceCapability), only one of them may have the specified field set to the identified unique value.	Change the value to a value that is not required to be unique, change the value of that field in the other record currently holding the unique value, or remove the field from the request. Note - this validation error response will additionally include a property "conflictObjectId", set to the OBJECTID of the existing record that is causing the conflict.

	Error querying for IrwinRID in ResourceCapability: Database error	Depending on the error, this may indicate a server failure of some kind.	Check your input data and try again. Depending on the error, this may indicate a server failure of some kind. If this happens repeatedly, please report it to the implementation team.
305	Unable to validate whether value is unique because the value of 'NameMiddle' could not be parsed.	Uniqueness for records within a given layer may be enforced based on one or more fields; at least one of those fields couldn’t be read correctly.	Ensure all required fields are present in the request and conform to the respective documented requirements.
	values (NameFirst, NameLast, NameMiddle, HomeDispatchUnit, ProviderUnit, BirthMonthDay) must be unique in Resource (conflicts with IrwinRID ‘069BA152-C519-4502-A560-3F72754FB862’).	Uniqueness for records within a given layer may be enforced based on one or more fields; the aggregate value of those fields in the submitted request would result in a duplicate (non-unique) record.	Change at least one of the values identified in the message to ensure the combination of those fields is unique. Note - this validation error response will additionally include a property “conflictObjectId”, set to the OBJECTID of the existing record that is causing the conflict.
	Error querying for LastName in Resource: Database error	Depending on the error, this may indicate a server failure of some kind.	Check your input data and try again. Depending on the error, this may indicate a server failure of some kind. If this happens repeatedly, please report it to the implementation team.
306	Unable to validate whether value is valid because the value of 'ResourceKind' could not be parsed.	Some properties are only valid for certain ResourceKinds; the value of ResourceKind couldn’t be determined so there is no way to determine if the specified property is valid.	Ensure the value of ResourceKind is included in the request and is valid.
	Unable to validate whether value is valid because the resource kind could not be determined(1).		

	Unable to validate whether value is valid because the resource kind could not be determined(2).		
	Unable to validate whether value is valid because the resource kind could not be determined(3).		
	a value is not allowed for resource kind Overhead.	The property specified is not valid for the ResourceKind specified (in this case, Overhead).	Remove the value from the request.
307	Unable to validate whether value is required because the value of 'ResourceKind' could not be parsed.	Some properties are only required for certain ResourceKinds; the value of ResourceKind couldn't be determined so there is no way to determine if the specified property is required.	Ensure the value of ResourceKind is included in the request and is valid.
	Unable to validate whether value is required because the resource kind could not be determined(1).		
	Unable to validate whether value is required because the resource kind could not be determined(2).		
	Unable to validate whether value is required because the resource kind could not be determined(3).		
	a value is required for resource kind Overhead.	The property specified must have a non-null value for the ResourceKind specified (in this case, Overhead).	Ensure the property is included in the request and has a valid, non-null value.
308	value must be exactly 17 characters long.	VINs must be exactly 17 characters long.	Ensure the VIN conforms to the standard defined in the US CFR.

	value cannot contain the letters 'I', 'O', or 'Q'.	A VIN may not contain any of the letters "I" (9th letter of the English alphabet), "O" (15th), or "Q" (17th).	
	Not a valid VIN.	The specified VIN does not conform to the standard defined in the US CFR.	

5 API Operations

The Resource API operations can be accessed at the following link. The *SWAGGERhub* documentation provides an interactive platform for both understanding method specifications and also testing interacting with the methods. Note that this documentation tool is connected to the IRWIN TEST environment only**.**

https://app.swaggerhub.com/apis-docs/IRWIN7/Irwin/6.0.0

https://app.swaggerhub.com/apis-docs/IRWIN7/Resource/6.0.0