July 3, 2009
Candidate Standard 5108.03-2009

The Printer Working Group

Network Resource Service Semantic Model and Service Interface

Version 1.0
Status: Approved

Abstract: When network Multifunction Devices are installed in local office or enterprise networks shared by large groups of users, the ability to provide resources such as job tickets pre-configured with user's intent (Job Resource), professionally prepared Logos, Fonts, Forms, …, etc, that can be reused by user's jobs is very important for office document productivity. This specification defines a Resource Service that provides operators and users a convenient way to remotely store, manage resources so that they can be retrieved and shared later through job creation requests to other services of network Multifunction Devices.

This document is a PWG Candidate Standard. For a definition of a "PWG Candidate Standard", see:
ftp://ftp.pwg.org/pub/pwg/general/pwg-process20.pdf
This document is available electronically at:
ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm20-resource10-20090703-5108.03.pdf, .doc

Title: Network Resource Service Semantic Model and Service Interface

The IEEE-ISTO and the Printer Working Group DISCLAIM ANY AND ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The Printer Working Group, a program of the IEEE-ISTO, reserves the right to make changes to the document without further notice. The document may be updated, replaced or made obsolete by other documents at any time.

The IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO take no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights.

The IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO invite any interested party to bring to its attention any copyrights, patents, or patent applications, or other proprietary rights, which may cover technology that may be required to implement the contents of this document. The IEEE-ISTO and its programs shall not be responsible for identifying patents for which a license may be required by a document and/or IEEE-ISTO Industry Group Standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. Inquiries may be submitted to the IEEE-ISTO by e-mail at:

info@ieee-isto.org

The Printer Working Group acknowledges that the IEEE-ISTO (acting itself or through its designees) is, and shall at all times, be the sole entity that may authorize the use of certification marks, trademarks, or other special designations to indicate compliance with these materials.

Use of this document is wholly voluntary. The existence of this document does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to its scope.

About the IEEE-ISTO

The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and flexible operational forum and support services. The IEEE Industry Standards and Technology Organization member organizations include printer manufacturers, print server developers, operating system providers, network operating systems providers, network connectivity vendors, and print management application developers. The IEEE-ISTO provides a forum not only to develop standards, but also to facilitate activities that support the implementation and acceptance of standards in the marketplace. The organization is affiliated with the IEEE (http://www.ieee.org/) and the IEEE Standards Association (http://standards.ieee.org/).

For additional information regarding the IEEE-ISTO and its industry programs visit:
http://www.ieee-isto.org.

About the Printer Working Group

The Printer Working Group (or PWG) is a Program of the IEEE-ISTO. All references to the PWG in this document implicitly mean “The Printer Working Group, a Program of the IEEE ISTO.” The PWG is chartered to make printers and the applications and operating systems supporting them work together better. In order to meet this objective, the PWG will document the results of their work as open standards that define print related protocols, interfaces, data models, procedures and conventions. Printer manufacturers and vendors of printer related software would benefit from the interoperability provided by voluntary conformance to these standards.

In general, a PWG standard is a specification that is stable, well understood, and is technically competent, has multiple, independent and interoperable implementations with substantial operational experience, and enjoys significant public support.

Contact information:

The Printer Working Group
c/o The IEEE Industry Standards and Technology Organization
445 Hoes Lane
Piscataway, NJ 08854
USA

MFD Web Page: http://www.pwg.org/mfd MFD Mailing List: mfd@pwg.org

Instructions for subscribing to the MFD mailing list can be found at the following link:
http://www.pwg.org/mailhelp.html
Members of the PWG and interested parties are encouraged to join the PWG and MFD WG mailing lists in order to participate in discussions, clarifications and review of the WG product.

TABLE OF CONTENTS

Figures

1 Introduction

This document specifies the PWG abstract model for a Resource Service of a Multifunction Device (MFD). Included in this document are the content specific terminology, data model, the theory of operation, the resource service interfaces and the conformance requirements. The MFD resource service abstract models include the functional models and interfaces of the associated resource services for a multifunction device.

2 Summary

Resource Service allows professionally prepared job processing resources to be stored and then reused later for repetitive job processing, sharing by users, and management in centralized or distributed manner for network MFDs.

A network MFD client application has a client component for each of its targeted MFD services and optionally a Resource Client. A network Resource Client application interacts with the user to retrieve a pre-stored Resource, such as those listed below in order to create or process a job:

- a Template containing the pre-set descriptive and processing parameters of a job or document suitable for submission to a targeted MFD service. A Template contains instructions representing the user’s preconfigured intent that can be used as-is or modified by the user, if authorized. Once the user is satisfied with the Template, the network Template Client application passes the Template to an intended MFD Service Client for creating job tickets or document tickets. These tickets are then used for job creation by the Service .
- an input or output ICC Profile enabling correct color space conversion
- a Form for document production for insertion at the front page of a document
- an image (e.g. Watermark, Logo, background) for Overlays
- a Font for text printing

The Resource Service in this specification provides the abstract data model of the service and the operations for retrieval, storage, replacement, deletion, listing Resources, and obtaining summary information of Resources and Resource Services available to network MFD Services. Like all other MFD services, a Resource Service can be independently started, shutdown, disabled, or enabled by an authorized administrator at any time appropriate for the service.

3 Terminology

3.1 Conformance Terminology

Capitalized terms, such as **MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, RECOMMENDED** and **OPTIONAL**, have special meaning relating to conformance as defined in [RFC2119].

MUST This word means that the definition is an absolute requirement of the specification.

REQUIRED This word means that the definition is an absolute requirement of the specification.

SHALL This word means that the definition is an absolute requirement of the specification.

MUST NOT This phrase means that the definition is an absolute prohibition of the specification.

SHALL NOT This phrase means that the definition is an absolute prohibition of the specification.

SHOULD This word means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

RECOMMENDED This word means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

NOT RECOMMENDED This phrase means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

OPTIONAL This word means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

3.2 Content Specific Terminology

Term **Definition**

Attribute Attributes can decorate XML Elements and contain additional information about an Element.

Document Data The digitized data submitted by an user as the document to be processed by an MFD service, or as the resulting data from the scanning of Hardcopy Document(s) in an MFD. The images from the scanned Hardcopy Document(s) are encoded in an image or document format and stored at a Destination. Document Data is not considered as a resource that can be retrieved or stored by the Resource Service.

Document Ticket A data object that contains user's intent for document processing and descriptive properties of a document within a job of a Service. The content of a Document Ticket is configured by an user through a network MFD Client application.

Element Elements are used to convey structure and relationships in XML document instances. An Element can contain both content and Attributes.

Term **Definition**

Executable Resource An executable code that is installed in an MFD system and executed for performing a task. Executable Resource includes two types of resources: Firmware, and Software. (See Firmware, Software definitions below.) Executable resource is a category of resources that is served by the Resource Service.

Firmware Persistent computer instructions and data embedded in the MFD that provides the basic functions of that device. Firmware is only replaced during a specialized update process. [IEEE2600]

Font A complete character set of a single size and style of a particular typeface. Mostly today computer fonts are based on fully scalable outlines. However, it still refers to a single style. Times New Roman regular, italic, bold and bold italic are four fonts, but one typeface. Font is a type of Static Resource that can be retrieved and stored by PWG MFD Resource Services.

Form A document (printed or electronic) with spaces in which to write or enter data. Form in this document refers to an electronic form, which is a type of Static Resource that can be retrieved and stored by PWG MFD Resource Services.

Group Element A collection of Elements that constitutes a complex Element.

ICC Profile A set of data that characterizes a color input or output device, or a color space, according to standards promulgated by the International Color Consortium (ICC). Profiles describe the color attributes of a particular device or viewing requirement by defining a mapping between the device source or target color space and a *profile connection space* (PCS) (see definition below). This PCS is either CIELAB (L*a*b*) or CIEXYZ. Mappings may be specified using tables, to which interpolation is applied, or through a series of parameters for transformations. Every device that captures or displays color can have its own profile. ICC profile is a type of Static Resource that can be retrieved and stored by a PWG MFD Resource Service.

Job Ticket A data object that contains the user's intent for job-level document processing, job processing and descriptive properties of a job of a Service. The content of a Job Ticket is configured by user through a network MFD Client application.

MFD Service A software service that accepts and processes requests to create, monitor and manage Jobs requesting a service. The software service accepts and processes requests to monitor and control the status of the service itself and its associated resources. An MFD Service is hosted either locally or remotely to the MFD.

Local Client A software application entity that is co-located with the service it interacts with on behalf of an user.

Logo A graphical element, (ideogram, symbol, emblem, icon, sign) that, together with its **logotype** (a uniquely set and arranged typeface) form a trademark or commercial brand. Logo is a type of Static Resource that can be retrieved and stored by a PWG MFD Resource Service.

Profile Connection Space (PCS) A standard device independent color space defined by the International Color Consortium (ICC) that is used for mapping the color space of one device to the color space of another by describing

Term **Definition**

the relationship of each device’s color space to this device independent color space.

Resource An data object that can be served by the Resource Service when required by an MFD system for performing a task or a job. There are two categories of resources required by an MFD: Executable Resource, and Static Resource. (See the definitions of Executable Resource and Static Resource).

Resource Client The local or remote software entity that interfaces with the user and interacts with a Resource Service.

Resource Service A software service that provides the interfaces for storing, retrieving, and maintaining users’ Resources.

Static Resource A non-executable electronic data object that is required by an MFD system for performing a task. Static Resource includes the following types of resource: Font, Form, Images, Logo, Template.

Sequence A type of XML structure that represents an ordered list of elements.

Software Persistent computer instructions and data placed on the MFD, via download, that are separate from, and not a part of, the base Firmware. Applications are an addition to the base Firmware that provides additional function beyond that provided by the base Firmware. Software is a type of Executable Resource.

Template A data object that contains descriptive information and the pre-configured content of a job ticket or document ticket for a specific MFD service. A template is not bound to a specific job or document. It can be stored or retrieved from a Resource Service, collocated on the MFD or hosted on a remote system. Template is a type of Static Resource that can be retrieved and stored by a PWG MFD Resource Service.

Watermark A recognizable image or pattern when printed on paper used to identify authenticity. Watermark is a type of Static Resource that can be retrieved and stored by a PWG MFD Resource Service.

4 Requirements

4.1 Rationale for Resource Service Specification

There are clear benefits to provide a Resource Service for MFD Services. If MFD resources can be professionally prepared, stored and then retrieved using a standard, interoperable interface, then many MFD applications can be developed and flexibly integrated to reuse and share, or manage the resources centrally or in distributed manner across all MFDs in an enterprise network environment.

In order to implement an abstract model of the operations and elements for Resource Services, there is need to map them onto implementable applications and communication protocols that support interactions between Resource Clients and Resource Services. There is a clear need to define a binding of the abstract model into Web Service Schema and Web Service protocol stack.

4.2 Out of Scope for Resource Service Model

A Resource can be used in many ways to benefit users and their organizations. The purpose of the Resource Service specified herein is to provide users a convenient and interoperable method to store, retrieve, and manage resources for the jobs of MFD Services. The following are outside the scope of this model:

(1) The mechanism used for protection of Resource from unauthorized access, disclosure, and alteration.
(2) The metadata describing a resource storage location.
(3) Operations required to store and retrieve a Resource to and from a resource storage location.
(4) Document data submitted by an user to be processed by a job-oriented MFD service, or as the resulting data from the scanning of Hardcopy Document(s).
(5) The management of the site policy for the use of Resource Service features
(6) Digital signing and encryption methods for Resources

4.3 Model mapping conventions

The Resource Service model is described in this document as an XML Schema. This is for the sake of convenience and does not require a protocol mapping involving XML. The top level MFD objects, such as the subunits, the services, and their associated Jobs and Documents, can be represented in any number of ways. Abstractly they are objects which contain attributes or properties that express characteristics of the object. For the remainder of this document references to Attribute or Element refer to XML Attributes and XML Elements respectively. Either of these can be abstractly considered to be attributes or properties of abstract objects. In the data model diagrams an element in a solid box indicates a mandatory element. An element in a dashed box indicates an optional element. The pale yellow boxes indicate data types that are specific to the XML Schema encoding and are not relevant to the abstract model.

5 Resource Service Concept Diagram

The diagram below depicts the top level architecture of Resource Service. A Resource Service may reside locally within a MFD system or on a remote system. It provides resource storage and retrieval interfaces to a resource client object which may also reside locally within a MFD system or on a remote system. A user always requests resource storage or retrieval by interacting with a Local Resource Client or a Remote Resource Client. A Remote Resource Client may connect to multiple Resource Services at any period of time. By the same token, a Resource Service may serve multiple Remote Resource Clients. Each Resource Service may process zero or more resource storage and retrieval requests simultaneously. A Resource Service stores or retrieves a resource at or from a resource storage location. For security consideration, the resource storage location is not exposed to users, must be only configurable by an authorized administrator, and may be local or remote to the Resource Service.

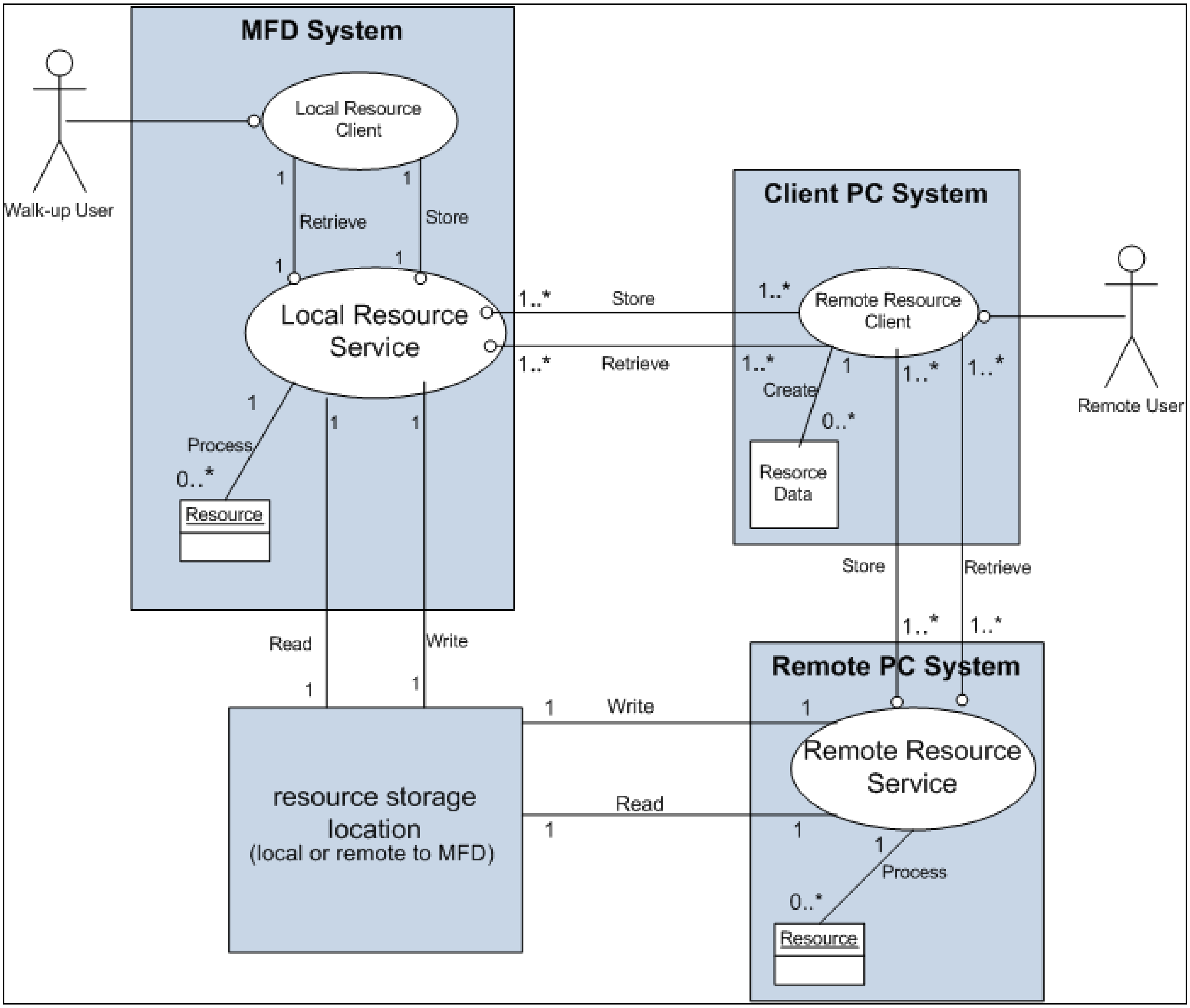


Figure 1 Resource Service Top Level Architecture

The lifecycle of a Resource begins when the Resource is created and ends when the Resource is deleted.

An MFD resource is created by an MFD Resource Client, then stored by a Resource Service at a resource storage location. The MFD Resource Client, Resource Service, or the resource storage location is local or remote to the MFD System.

At any time a resource may encounter the following events while stored in at resource storage location:

(1) retrieval by a Resource Service at the request from a Resource Client invoked by an MFD application on behalf of a user's job processing request,
(2) update/replacement by a Resource Service at the request of a Resource Client on behalf of the creator of the resource,

(3) expiration that signals the end of life for the resource, and the resource will then be deleted by the Resource Services.

6 Resource Service Model Overview

Resource Service is an optional Imaging Service. There may be zero or multiple instances of Resource Services in an Imaging System.

The types of resources within the scope of Resource Service include, but are not limited to: Font, Form, Logo, ICC Profile, Firmware, Software, Image, Template (including job Template and document Template).

The Resource Service model contains several service properties: service capability, service configuration, service description, service status, and an extension point for vendor's unique service properties.

The service capability of Resource Service has a ResourceDescriptionCapabilites element that contains the abilities of the service in supporting some of the descriptive attributes of a resource entry. These descriptive attributes of a resource entry include natural language for any service generated message or description text, date and time of expiration, Resource category, Resource creator user name, Resource format, Resource descriptive information, Resource name, and Resource type. A Resource Service does not provide other job-related service capabilities because it only stores and retrieves Resources, does not process any MFD job. The service capability of Resource Service also has an extension point for other vendor's unique service capabilities.

The service configuration is the Resource Service specific view of the applicable MFD subunits which may include Storages, Interfaces, InputChannels, and OutputChannels.

The service description property contains only the basic set of imaging service description elements common to all imaging services. It includes owner's web address and identification information, service URIs, service descriptive information, service location, and service name.

The service status property contains state information maintained by the service automata. The state information consists of a set of basic imaging service states which are common to all imaging services and a set of service states that are Resource Service specific.

The basic imaging service state information includes service access rights, service creation date, current time, service id, a flag indicating whether the service is accepting jobs (here refers to requests for storing and retrieving resources in Resource Service), the natural language currently used by the service, queued service request count, serial number of the device hosting the service, the current state of the service, the current state of the subunits used by the service, additional information about the service state, and human readable messages on the additional information about the service state.

The Resource Service specific set of service state contains information on resource service counters that provide Resource Service specific usage accounting information.

Below is the top level view of the Resource Service schema.

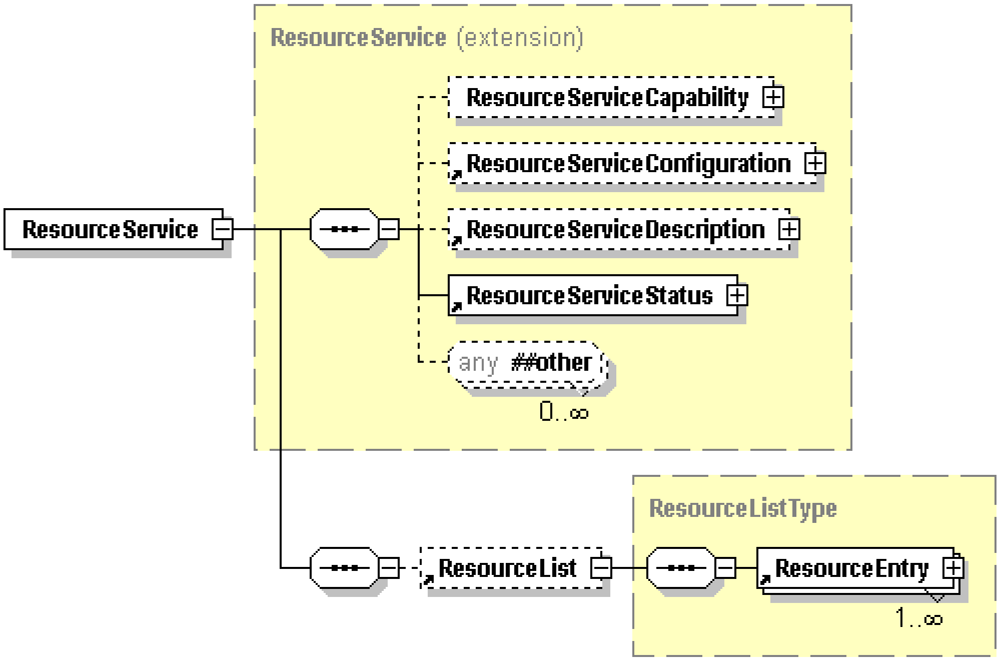

ResourceService (extension)
ResourceServiceCapability
ResourceServiceConfiguration
ResourceServiceDescription
ResourceServiceStatus
any ##other
0..∞
ResourceService
ResourceListType
ResourceList
ResourceEntry
1..∞

Figure 2 High Level Resource Service Schema

Besides service properties, Resource Service has an unordered list of Resources: ResourceList. Each element in the list is a ResourceEntry that contains Resource descriptive and status information for the stored resource.

Resource Service provides a set of client interfaces that allow users to list all resources available to a specific user or group, retrieve a resource that is specified by the Resource ID, store a new resource that validates a resource before it is stored, replace (update) and delete a resource with the specified Resource ID, and get and set the metadata of a Resource and a Resource Service.

Like all other imaging services, Resource Service model provides a set of administrative interfaces that allows authorized administrators to remotely disable/enable the service, and shutdown/restart the service.

7 Resource Service Model Description

7.1 Data Types

The keywords used to define the data types of the elements of the Resource Service model are the same as those defined in the *Network Scan Service Semantic Model and Service Interfaces* specification [PWG 5108.02-2009].

7.2 Resource Service Configuration

(complex-ResourceServiceConfiguration) This group element identifies the actual instances of the subunits the Resource Service uses. These elements are based on the Printer MIB [RFC3805]. See the RFC for details on these elements.

Note that not all subunits are used by the Resource Services.

The following diagram shows the schema of Resource Service Configuration.

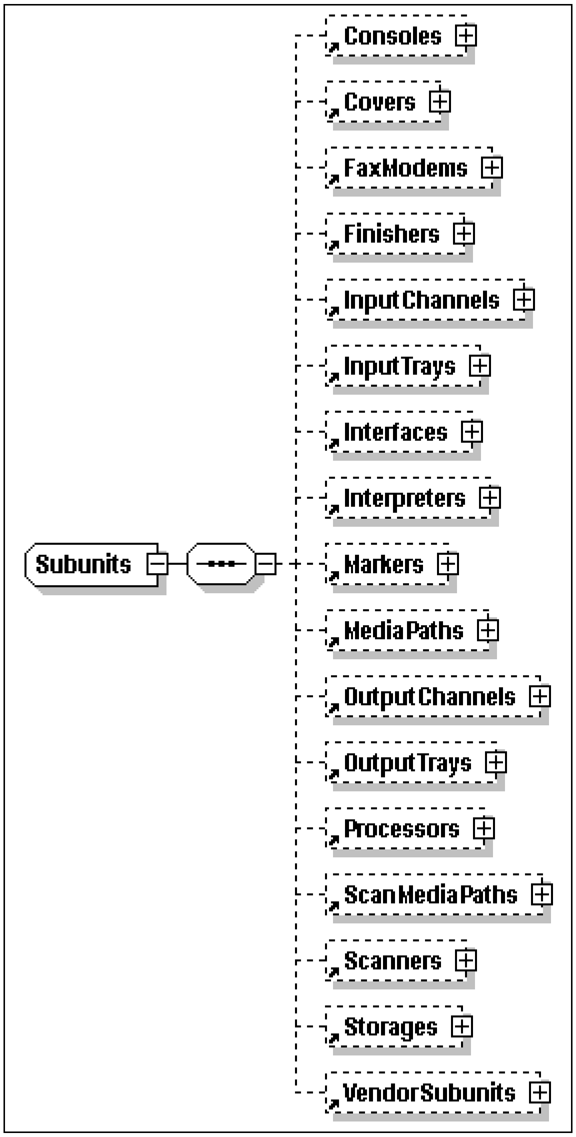


Figure 3 Resource Service Configuration Schema

All subunits defined for all imaging services are optional for a Resource Service. The subunits applicable to Resource Service are Consoles, Covers, Interfaces, InputChannels, OutputChannels, Storages, and VendorSubunits.

For the descriptions of the subunits Consoles, Covers, Interfaces, InputChannels, OutputChannels, and VendorSubunits please refer to the PWG Candidate Standard Network Scan Service Semantic Model and Service Interface Version 1.0 [PWG5108.02-2009].

The subsections below describe the Storage subunit that is specific to the Resource Service.

7.2.1 Storage

(complex-Storage) Each storage subunit contains a StorageStatus element and a StorageDescription element.

Below is the XML Schema diagram of the Storage subunit.

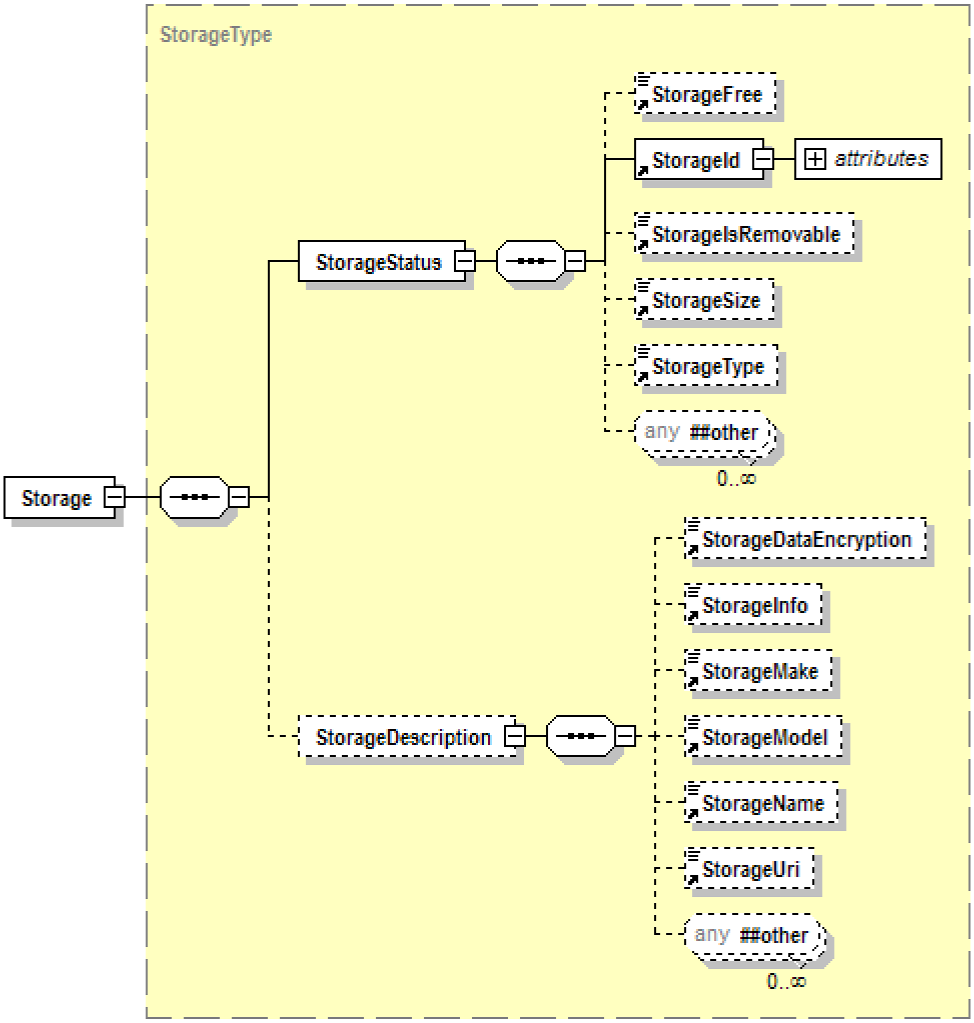


Figure 4 Storage Subunit Schema

A Storage contains the elements described in the following sections.

7.2.1.1 Storage Status

(complex-StorageStatus) This REQUIRED group element contains the current status of the storage subunit.

7.2.1.1.1 StorageId

(int) This REQUIRED element is the unique identifier of the storage subunit within the Resource Service instance.

7.2.1.1.2 StorageType

(keyword) This element indicates the type of the storage subunit. The allowed storage types are expanded from the IETF Host Resource MIB standard [RFC 2790] as in the following list:

RAM - Random access memory,
NVRam - Non-volatile random access memory, e.g. flash memory (non-removable flash), flash card (removable flash),
ramDisk - Random access memory used as hard disk for secondary storage,

HardDisk - Non-removable, rigid, rotating storage,
FloppyDisk - Non-rigid rotating magnetic storage,
VirtualMemory - Temporary storage of swapped or paged memory,
NetworkStorage - Any storage external to an MFD,
CD - Compact disk.
DVD - Digital versatile disk,
OpticalDiskROM - Optical disk, Read only memory,
OpticalDiskWORM - Write once read many optical disk,
OpticalDiskRW - Read writable optical disk,
Other (Any other vendor defined storage).

7.2.1.1.3 StorageSize

(int) This element indicates the total bytes allocated to the service, i.e. the service-specific view of storage size.

7.2.1.1.4 StorageFree

(int) This element indicates the total bytes currently free for use to the service.

7.2.1.1.5 StorageIsRemovable

(Boolean) This element indicates whether the Storage subunit is removable by the design of its manufacturer.

7.2.1.1.6 List of ##Other

This serves as the list of vendor's extension points for the attributes of StorageStatus.

7.2.1.2 Storage Description

(complex-StorageDescription) The group element provides descriptive information on the specific storage subunit used by the Resource Service.

7.2.1.2.1 StorageName

(string) This is the name of the storage subunit that is unique within a Resource Service instance.

7.2.1.2.2 StorageMake

(string) This element contains information on the manufacturer of the storage subunit.

7.2.1.2.3 StorageModel

(string) This element describes the model name of the storage subunit.

7.2.1.2.4 StorageInfo

(string) This element contains administrator settable additional description text about the storage subunit

7.2.1.2.5 StorageDataEncryption

(boolean) This element indicates the support of data encryption by the storage subunit

7.2.1.2.6 List of ##Other

This element provides a list of vendor's extension points for StorageDescription while maintaining interoperability.

7.2.1.2.7 StorageURI

(URI) This element is only applicable to a NetworkStorage device . It provides the network location information of the storage subunit.

7.3 Resource Service Description

(complex-ResourceServiceDescription) The ResourceServiceDescription group element provides descriptive information for the Resource Service. The constituent elements are administratively set. The element values once set can be modified directly or indirectly through an operation of a service or a management protocol such as WS-Management. Currently Resource Service does not define such administrative operations.

Note that Resource Service Description consists of two sequences. The first represents elements inherited from the Imaging Service super class and the second sequence includes Resource Service specific extensions.

Below is a view of the Description elements for the Resource Service.

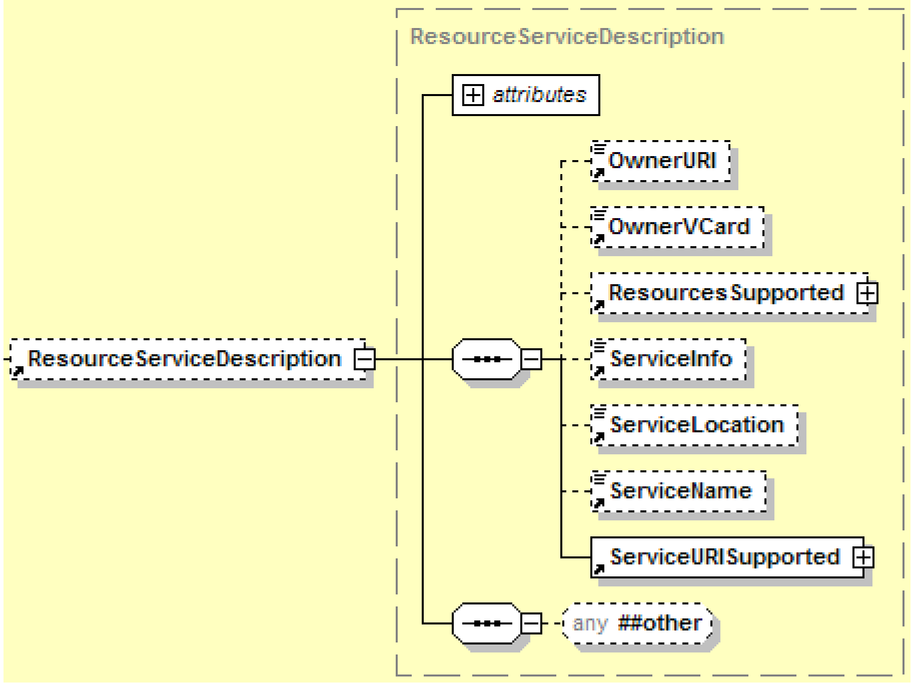


Figure 5 Resource Service Description Schema

7.3.1 OwnerURI

(URI) This is the URI by which you can reach the administrator or owner who created the service.

7.3.2 OwnerVCard

(string) This is the vCard [RFC2426] of the owner of the Resource Service.

7.3.3 ResourcesSupported

(list of complex-AvailableResource) This attribute is not applicable to Resource Service. This is a service specific view of the available resources used by a service. Note that the AvailableResources element at the MFD server's top level contains installed resources on a specific MFD, not physical resources to be managed by Resource Service.

7.3.4 ServiceInfo

(string) This is descriptive information about this Resource Service the owner of the service wishes to provide.

7.3.5 ServiceLocation

(string) This element describes the physical location of this Service. (Example: "Building 128 Floor 2 Room 210C")

7.3.6 ServiceName

(string) This is the end-user friendly name of the Resource Service.

7.3.7 ServiceURISupported

(list of URI) This REQUIRED element contains the URIs where the Resource Service is available.

7.3.8 Sequence of ##other

(list of any) This optional element contains extension points for vendor differentiation and implementation specific extensions while maintaining interoperability.

7.3.9 Attribute of ##other

(any) This is a REQUIRED vendor's extension point for adding one single extra attribute to Resource description properties.

7.4 Resource Service Capability

(complex-ResourceServiceCapability) This element contains the ResourceDescriptionCapabilities group element which specifies the descriptive elements supported in the ResourceDescription of each ResourceEntry.

Below is the Schema diagram of ResourceServiceCapability.

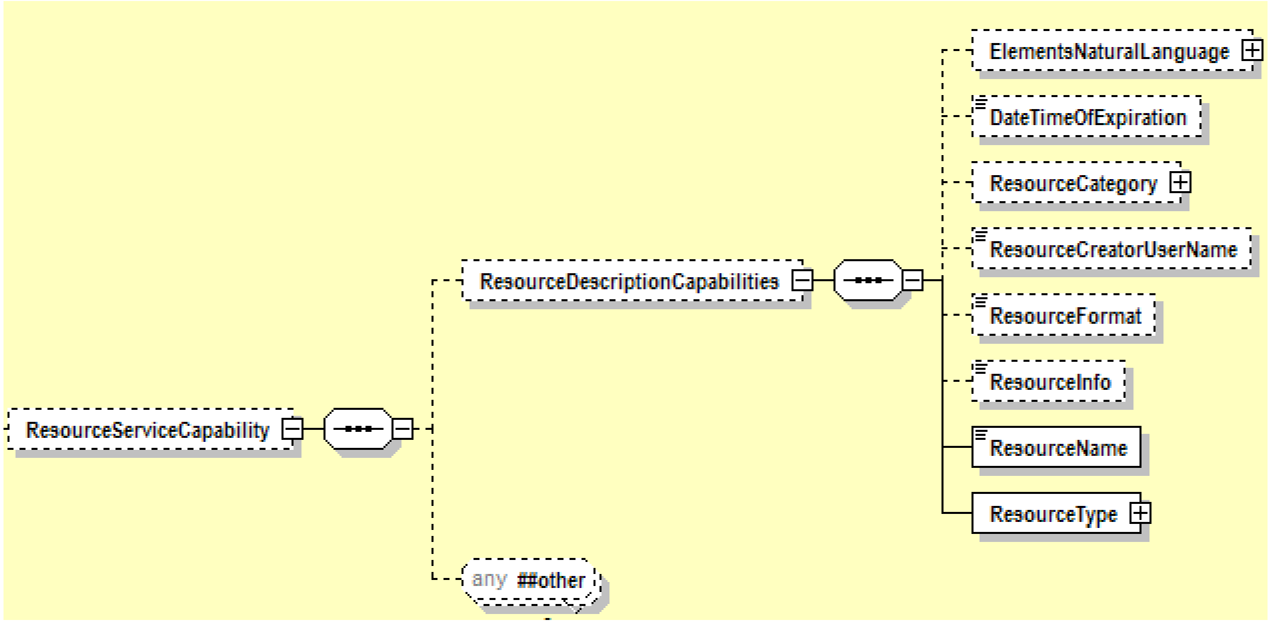


Figure 6 Resource Service Capability Schema

7.4.1 List of ##other

(list of any) This optional list contains extension points for vendor differentiation and implementation specific extensions while maintaining interoperability.

7.4.2 Resource Description Capabilities

(complex-ResourceDescriptionCapabilities) This is a group of elements specifying the ResourceDescription elements (see section 7.6.1.1) supported by the service.

7.4.2.1 DateTimeOfExpiration

(boolean) This Boolean valued element indicates whether the DateTimeOfExpiration element of ResourceDescription is supported by the service.

7.4.2.2 ElementsNaturalLanguage

(list of keyword) This element contains a sequence of AllowedValues that specify the supported natural languages for the ResourceDescription elements with a string syntax (example values: en-us, fr, de, ja) See [RFC4646]

7.4.2.3 ResourceCategory

(list of keyword) This element contains a sequence of AllowedValues that specify the supported keywords of the ResourceCategory element of the ResourceDescription. See section 7.6.1.1.1 for example values of supported ResourceCategory.

7.4.2.4 ResourceCreatorUserName

(boolean) This element indicates the support of the ResourceCreatorUserName element of the ResourceDescription by the service.

7.4.2.5 ResourceFormat

(boolean) This element indicates the support of the ResourceFormat element of the ResourceDescription.

7.4.2.6 ResourceInfo

(boolean) This element indicates the support of the ResourceInfo element of the ResourceDescription.

7.4.2.7 ResourceName

(boolean) This element indicates the support of the ResourceName element of the ResourceDescription.

7.4.2.8 ResourceType

(list of keyword) This element contains a sequence of AllowedValues that specify the supported values of the ResourceType element of the ResourceDescription. See section 7.6.1.1.4 for example values of supported ResourceType.

7.4.2.9 List of ##Other

This list contains an extension point for vendor differentiation and implementation specific extensions while maintaining interoperability.

7.5 Resource Service Status

(complex-ResourceServiceStatus) The Status elements provide state information of the Resource Service. The states are maintained by automata and can not be directly set.

Note that ResourceServiceStatus consists of two sequences. The first represents elements inherited from the Imaging Service super class and includes elements such as Id and State. The second sequence includes Resource Service specific extensions to the super class such as the Resource Service counters.

Below is a view of the Status elements for the Resource Service.

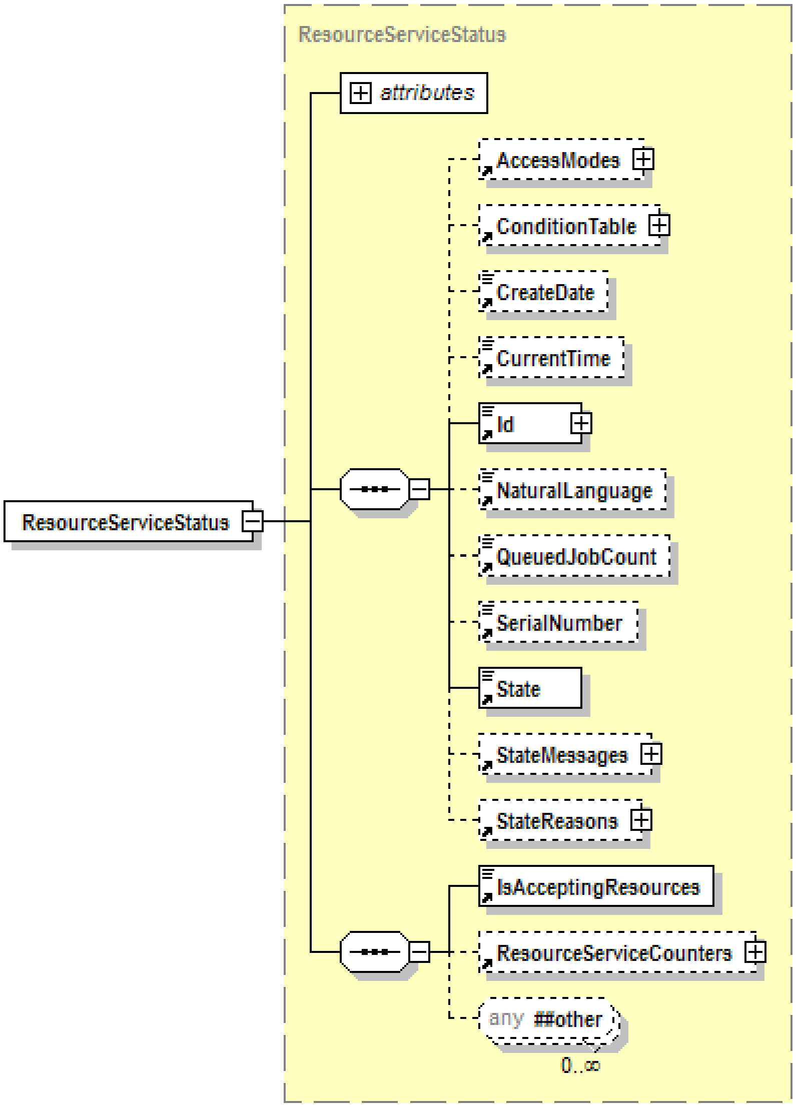


Figure 7 Resource Service Status Schema

7.5.1 AccessModes

(list of keyword) This element corresponds to the access mode property of a POSIX file and specifies the basic access control policy for the Resource Service object, as set by the Owner. The AccessMode element takes precedence over any externally specified access policy; e.g., Access Control Lists (ACLs) MUST NOT reduce the security required by this AccessMode element but MAY further restrict that access.

The AccessMode element describes the basic access control policy for the Owner, members of the Owner's group (e.g., Administrators), and all other authenticated users (i.e., the 'World'). Typically, only the Owner or a member of the Owner's group (site-specific) is granted the Write permission, but the World is granted Read permission (e.g., to read elements such as capabilities). [CHMOD, STAT]

7.5.2 CreateDate

(dateTime) Date and Time the Service was created.

7.5.3 CurrentTime

(dateTime) This element indicates the current date and time according to the Resource Services internal clock

7.5.4 ID

(int) A REQUIRED 32 bit Object Identifier for the Resource Service instance. It is unique within a Resource server.

7.5.5 IsAcceptingResources

(boolean) This element indicates whether this Resource Service is currently able to store or retrieve Resources. When the service is NOT accepting resources, the service MUST still be able to process requests for obtaining information on service configuration, service description, and service status. How the value for this element is configured is implementation-specific, e.g., local console, web page.

7.5.6 NaturalLanguage

(string) Indicates the local language currently used by the Service. This is used unless the operation specifies a RequestedNaturalLanguage .

7.5.7 QueuedJobCount

(int) This element does not apply to Resource Service. This element specifies the number of requests the service has currently accepted but not yet finished processing.

7.5.8 Resource Service Counters

(complex-ResourceServiceCounter) This element contains the counters for the amount of work performed by the Resource Service, timers covering utilization and monitoring information covering errors, warnings, traffic, resource counts and configuration changes. See PWG5106.1-2007.

7.5.9 SerialNumber

(string) This element specifies the serial number of the device hosting the service.

7.5.10 State

(keyword) This is an element of Resource Service Status. This REQUIRED element records the current state of the Resource Service instance. The state contains one of the following values:

From [RFC2911]:
Idle – The Resource Service is available and can start processing a new request.
Processing – The Resource Service is currently processing requests.

From [RFC2790]:
Unknown– The state of the Resource Service is not known.
Testing- The Resource Service is in testing or maintenance mode.
Down- The Resource Service is unavailable for service requests.

7.5.10.1 Service State Diagram

All imaging services inherit the same service state behavior.
The Service State Diagram is divided into three phases:

- <Init> - Unknown state - immediately after service creation
- <Offline> - Down and Testing states - no user requests are processed
- <Online> - Idle and Processing states

See '*Network Scan Service Semantic Model and Service Interface'* specification [PWG5108.02-2009] for the service state diagram.

7.5.10.2 Service State Transition Tables

The following notations are used in the two subsequent service state transition tables:

~ = logical NOT (e.g., ~C.critical means "not critical")
C = prefix of a condition
E = prefix of an event (e.g., E.endRequest means "request completed")
titlecase = state (e.g., Idle), operation (e.g., Startup), or phase
lowercase = action function (in FSM)
error = In Table 2 indicates that an error response is returned to the requestor
In Table 3 indicates that an error message should be logged indicating an implementation error

The following notes are used in the two service state transition tables:

(1) Startup and Restart
- Startup (Unknown/Init) sends E.startup and goes to (Down/Offline)
- Startup (Down/Offline) is a synonym for Restart
- Restart (Down/Offline) initializes and goes to (Idle/Online)

(2) Shutdown
- Shutdown (Testing|Idle) goes to (Down/Offline)
- Shutdown (Processing) sends E.shutdown and stays in (Processing)

(3) Testing
- Testing (Down) goes to (Testing/Offline)
- Testing (Idle|Processing) is an error

General Notes :

1) C.isAcceptingResources represents the ability of the service to retrieve, store, replace, and delete a resource (i.e. StoreResource will fail if ~C.isAcceptingResources). Other operations are not affected by this condition.

2) E.startRequest and E.endRequest refer to the event for starting and ending a resource transfer operation respectively. Resource transfer operations are StoreResource, RetrieveResource, ReplaceResource, and DeleteResource.

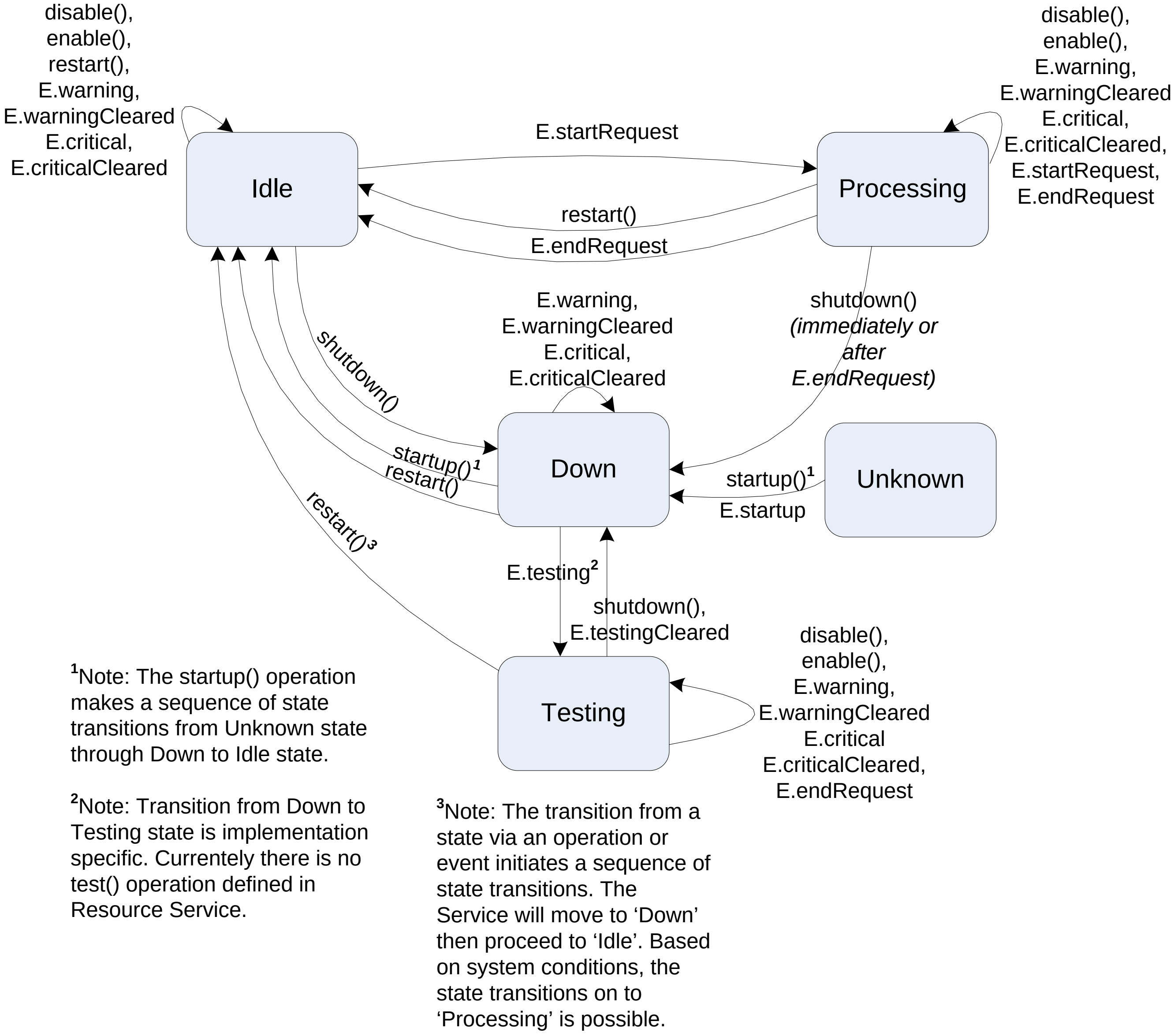
7.5.10.2.1 Service State Transition By Operations

SERVICE STATE MACHINE (Operations)				
	State			
Input	Down	Testing	Idle	Processing
Operation (Condition)	**Action (new state)**	**Action (new state)**	**Action (new state)**	**Action (new state)**
Disable	error	disable (~C.IsAcceptingResources)	disable (~C.IsAcceptingResources)	disable (~C.IsAcceptingResources)
Enable	error	enable (C.IsAcceptingResources)	enable (C.IsAcceptingResources)	enable (C.IsAcceptingResources)
Restart (Note 1)	restart (Idle)	restart (Idle)	restart (Idle)	restart (Idle)
Shutdown (Note 2)	error	shutdown (Down)	shutdown (Down)	shutdown (Down)
Startup (Note 1)	restart (Idle)	error	error	error
test (Note 3)	test (Testing)	test (Testing)	error	error

7.5.10.2.2 Service State Transition By Events

SERVICE STATE MACHINE (Events)				
	State			
Input	**Down**	**Testing**	**Idle**	**Processing**
Event (Condition)	**Action (new state)**	**Action (new state)**	**Action (new state)**	**Action (new state)**
E.critical	C.critical	C.critical	C.critical	C.critical
E.criticalCleared	~C.critical	~C.critical	~C.critical	~C.critical
E.endRequest (C.shutdown)	error	shutdown (down)	error	shutdown (Down)
E.endRequest	error	(Testing)	error	schedule (Idle or Processing)
E.warning	C.Warning	C.Warning	C.Warning	C.Warning
E.warningCleared	~C.Warning	~C.Warning	~C.Warning	~C.Warning
E.startRequest (~C.isAcceptingResources)	error	error (Testing)	error	error
E.startRequest	error	schedule (Testing)	schedule (Processing)	schedule
E.startup (Note 1)	restart (Idle)	error	error	error
E.testing (Note 3)	(Testing)	(Testing)	error	error
E.testingCleared	error	(Down)	error	error

7.5.10.2.3 Detailed Service State Transition Diagram



[1] Note: The startup() operation makes a sequence of state transitions from Unknown state through Down to Idle state.

[2] Note: Transition from Down to Testing state is implementation specific. Currentely there is no test() operation defined in Resource Service.

[3] Note: The transition from a state via an operation or event initiates a sequence of state transitions. The Service will move to ‘Down’ then proceed to ‘Idle’. Based on system conditions, the state transitions on to ‘Processing’ is possible.

Figure 8 Detailed Service Transition Diagram

7.5.10.1 State Messages

(complex-StateMessages) This element contains information about the Service State and StateReasons in human readable text. If the Service supports this element, it MUST be able to generate the messages in any of the natural languages supported by the Service.

7.5.10.2 StateReasons

(list of keyword) This element contains additional detail about the service's state. The standard keyword values are defined in §4.4.12 of [RFC2911]. Values of interest to a Resource Service include:

- AttentionRequired
- ConnectingToDevice
- Deactivated

- None
- Other
- StorageFull
- StorageAlmostFull
- Shutdown
- TimedOut

7.5.11 List of ##other

(list of any) This element contains an extension point for vendor differentiation and implementation specific extensions while maintaining interoperability

7.5.12 Condition Table

(complex-ConditionTable) The Condition Table provides additional information on the current and past state of Subunits used by Resource Service. The contents of the tables are based on the Alert Table from the Printer MIB.

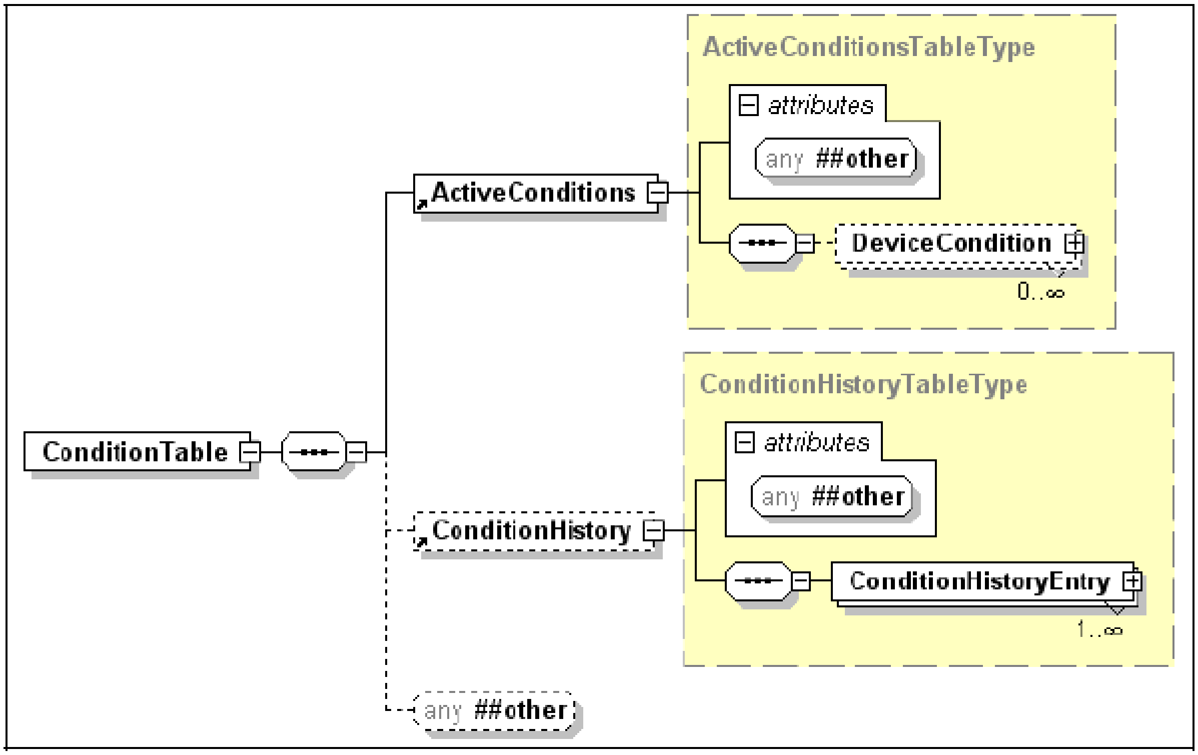


Figure 9 Condition Table

7.5.12.1 Active Conditions

(complex-ActiveConditions) This is a REQUIRED element for ConditionTable. This element represents the Resource Service specific view of conditions (informational, warning or critical) that are currently in effect on the device. This is modeled by a list of DeviceConditions and an extension point for a manufacturer specific condition.

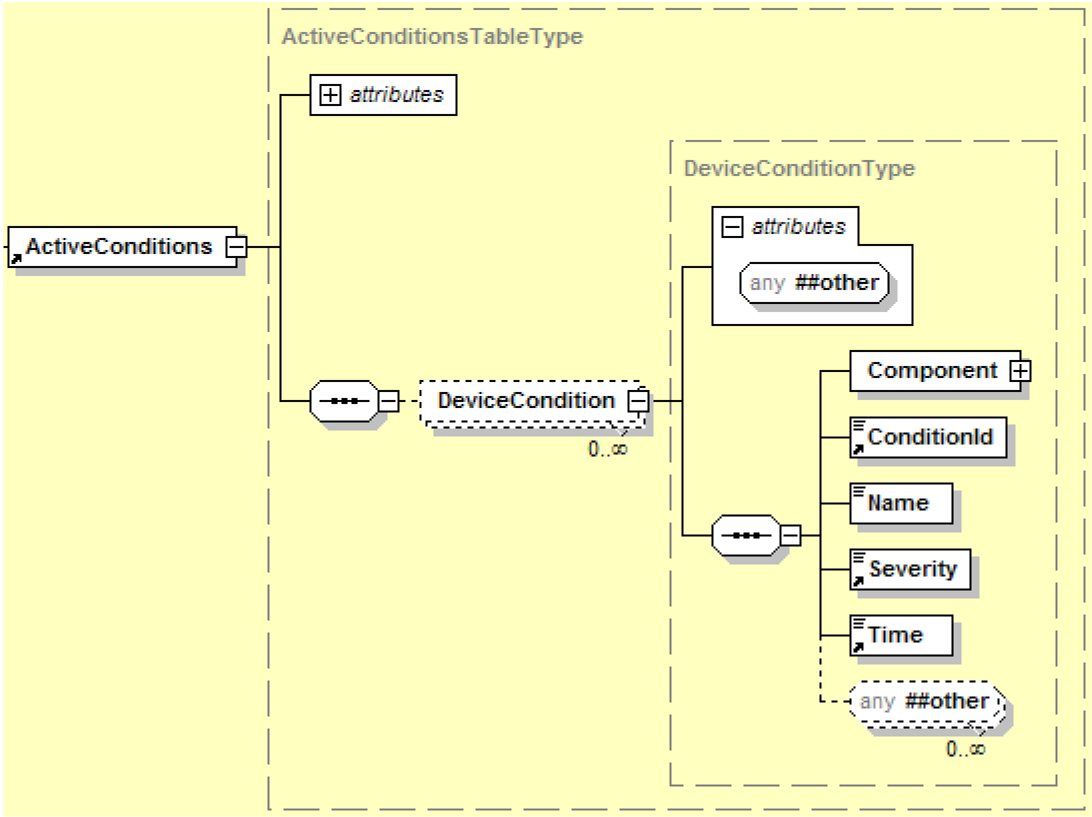


Figure 10 Active Conditions

7.5.12.1.1 Device Condition

(list of complex-DeviceCondition) This is an unbounded list of device conditions each contains the attributes described in the following subsections.

7.5.12.1.1.1 Component

(complex-Component) This element indicates the origin of the condition that is represented by the class and the name of the origin component.

7.5.12.1.1.1.1 Group

(keyword) This element specifies the class of the component (e.g. subunit, service) that is the source of the condition. Allowed values include: Storages, Console, Finisher, InputChannel, InputTray, Interface, Interpreter, Marker, MediaPath, OutputChannel, OutputTray, ScanMediaPath, Scanner, VendorSubunit, CopyService, EmailInService, EmailOutService, FaxInService, FaxOutService, NetFaxInService, NetFaxOutService, PrintService, ScanService, ResourceService, TransformService.VendorSubunit. The only values applicable to Resource Service are Storages and ResourceService.

7.5.12.1.1.1.2 Name

(string) This element specifies the instance of the component (e.g. subunit of resource service) that is the source of the condition. Values are taken from the Name element of the Group that is the source of the condition.

7.5.12.1.1.2 ConditionId

(int) This element is the unique identifier for the condition.

7.5.12.1.1.3 Name

(string) This element is the keyword that identifies the condition type. The values are the same values that are used for Service's StateReasons (e.g. AttentionRequired, StorageFull, StorageAlmostFull, …, etc.) .

7.5.12.1.1.4 Severity

(keyword) This element is the keyword that identifies criticality of the condition (i.e. Informational, Warning, Critical)

7.5.12.1.1.5 Time

(dateTime) This element is the date and time the condition occurred.

7.5.12.1.1.6 ##Other

(any) This is an extension point for manufacturer defined attributes for DeviceCondition.

7.5.12.2 Conditions History

(complex-ConditionHistory) This OPTIONAL table represents the conditions (informational, warning or critical) that have been cleared from the device. This element is an extension of the data type used by ActiveConditions. See above for the elements that are common. Only the ConditionHistory specific elements are explained below.

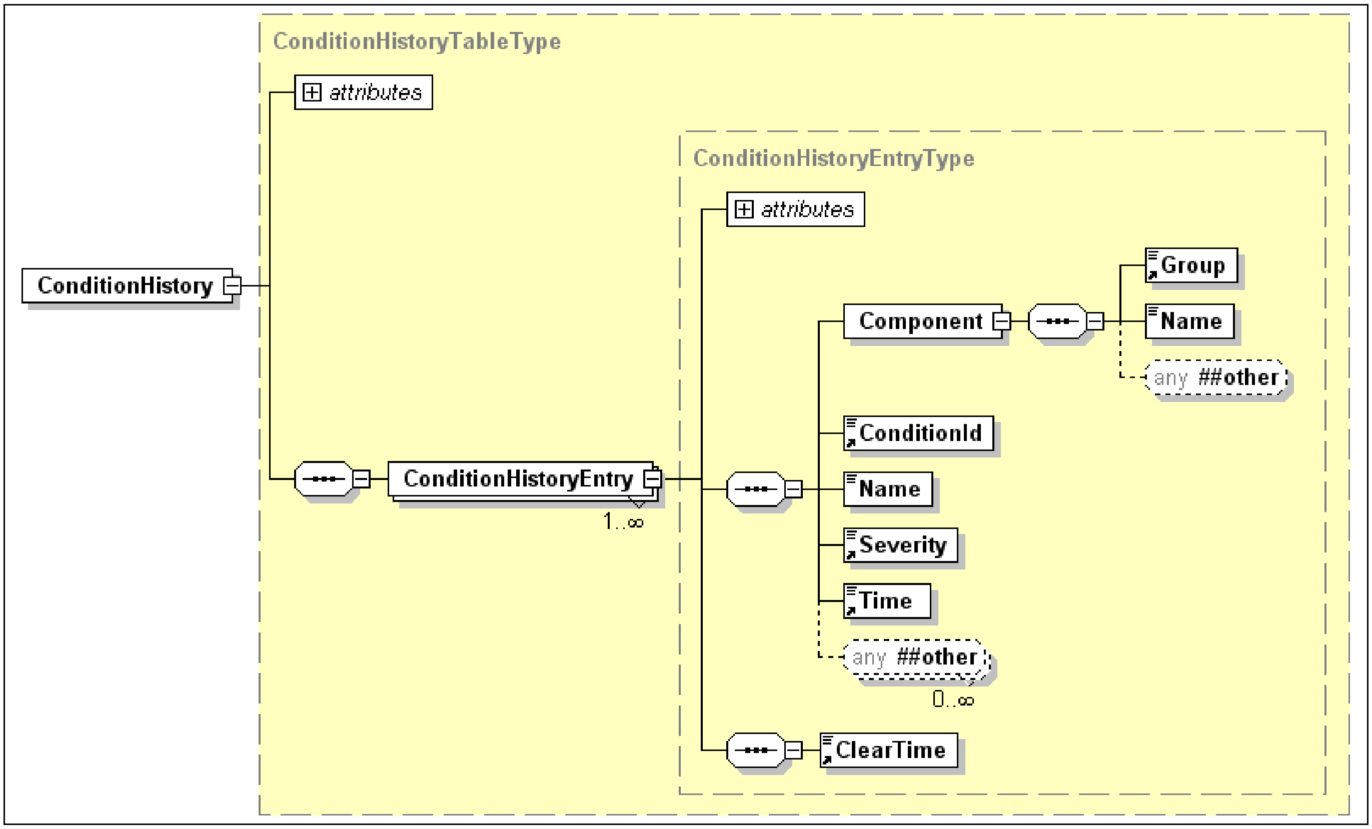


Figure 11 Condition History

7.5.12.2.1 ClearTime

(dateTime) This element is the date and time the condition was removed from the ActiveConditions list.

7.6 Resource List

(complex-ResourceList or list of complex-ResourceEntry) There is at least one entry of Resource in this list. This list records Resources on which a Resource Service operation described in section 8.1 has completed. There is no implied requirement of the ordering of the resources in the list. Below is the top level schema diagram of the ResourceList.

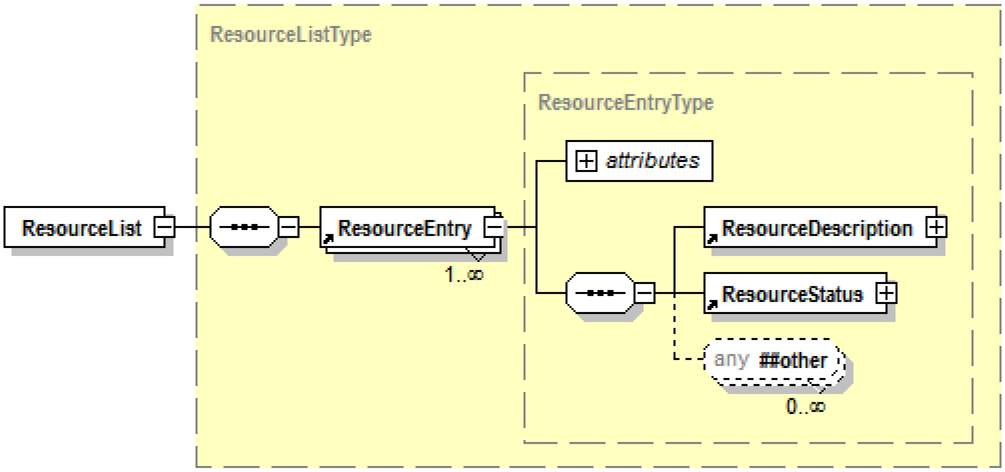
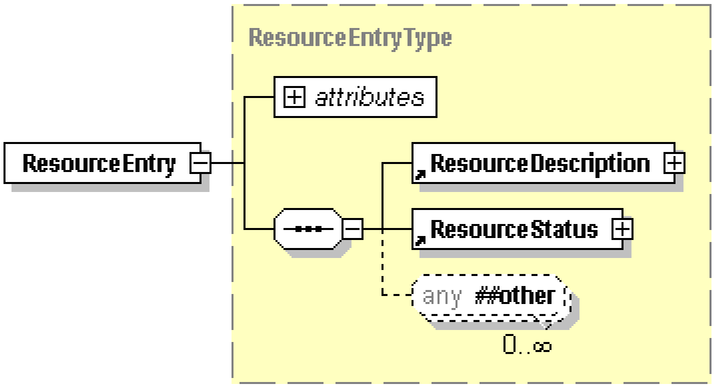


Figure 12 ResourceList Top Level Schema

7.6.1 Resource Entry

(complex-ResourceEntry) As depicted in Figure 10, each Resource Entry contains Resource Description, and Resource Status.



7.6.1.1 Resource Description

(complex-ResourceDescription) Resource Description is a REQUIRED group element of the Resource Entry.

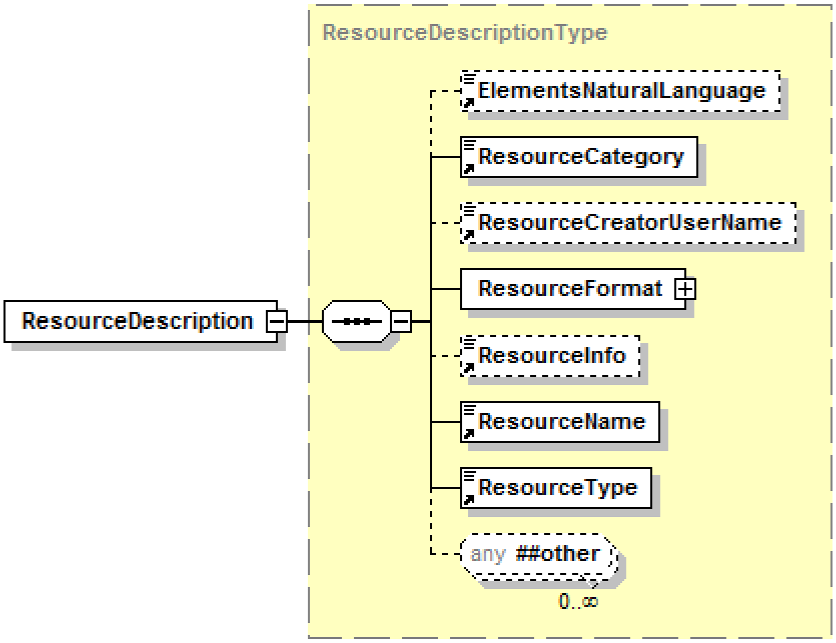


Figure 13 Resource Description Schema

7.6.1.1.1 ResourceCategory

(keyword) This REQUIRED element specifies the resource category of the ResourceEntry. The allowed values are: Executable, Static, Template, and Other.

7.6.1.1.2 ResourceCreatorUserName

(string) This is the user name unique within a network domain; the user of which created the resource. This element may be used to associate with the resource access control policy for the resource creator set by an authorized administrator.

7.6.1.1.3 ResourceName

(string) This REQUIRED element is the name of the Resource that uniquely identifies a Resource within a unique Resource Service instance.

7.6.1.1.4 ResourceType

(keyword) This REQUIRED element specifies the type of MFD Resources manageable by a Resource Service. The list of keywords allowed for ResourceType are the union of allowed resource types from three categories:

(1) Executable Resource category that allows one of the two types:
Firmware,
Software.

(2) Static Resource category that allowed the following types:
Font,
Form,
Image,
Logo,
ICCProfile.

(3) Template category that allows one of the following types:
CopyDocument, CopyJob,
EmailInDocument, EmailInJob,

EmailOutDocument, EmailOutJob,
FaxInDocument, FaxInJob,
FaxOutDocument, FaxOutJob,
NetFaxInDocument, NetFaxInJob,
NetFaxOutDocument, NetFaxOutJob,
PrintDocument, PrintJob,
ScanDocument, ScanJob,
TransformDocument, TransformJob.

Each template type indicates the resource is a job template or document template of a job ticket or document ticket for a particular job-oriented MFD Service. A job template or document template is a preconfigured job ticket or document ticket representing a user’s intent for a particular job or document respectively.

7.6.1.1.5 Resource Format

(complex-ResourceFormat) This REQUIRED element represents the format of the data for the resource. It is a choice of either DocumentFormat (string) which contains a MIME type value, or ResourceNamespace which is the URI string (XML namespace) identifying the resource data model (XML Schema).

DocumentFormat MUST be an enumeration of MIME type which is pwg:DocumentFormat type. (*sample values: application/octet-stream, application/pdf, application/postscript, application/vnd.pwg-xhtml-print+xml, application/vnd.hp-PCL, image/g3fax, image/jpeg, image/tiff, image/tiff-fx, text/html, text/plain, text/plain; charset=ISO-8859-1, text/plain; charset=US-ASCII, text/plain; charset=utf-8, unknown*). See [rfc2911] §3.2.1.1 and [PWG5100.5] §9.1.12.

7.6.1.1.6 ResourceInfo

(string) This element allows an user to provide arbitrary descriptive text for the resource.

7.6.1.1.7 ElementsNaturalLanguage

(keyword) This element indicates the natural language currently used to generate the descriptive text used by some elements in Resource Description.

7.6.1.1.8 List of ##other

(list of any) This list provides extension points for manufacturers to extend the Resource Description properties while maintaining interoperability.

7.6.1.2 Resource Status

(complex) Resource Status is another REQUIRED group element of the Resource Entry. A Resource is successfully stored, retrieved, deleted, or updated at the completion of a Resource operation. There is no intermediate status (or state), nor state reason associated with those intermediate states.

At the success of a Resource operation, a Resource ID MUST be created by the Resource Service to uniquely identify the Resource stored in or retrieved from a resource storage location.

ResourceStatus contains an element of any type (##other) for vendor-specific differentiation or implementation extension while maintaining interoperability.

Below is the Schema diagram of Resource Status.

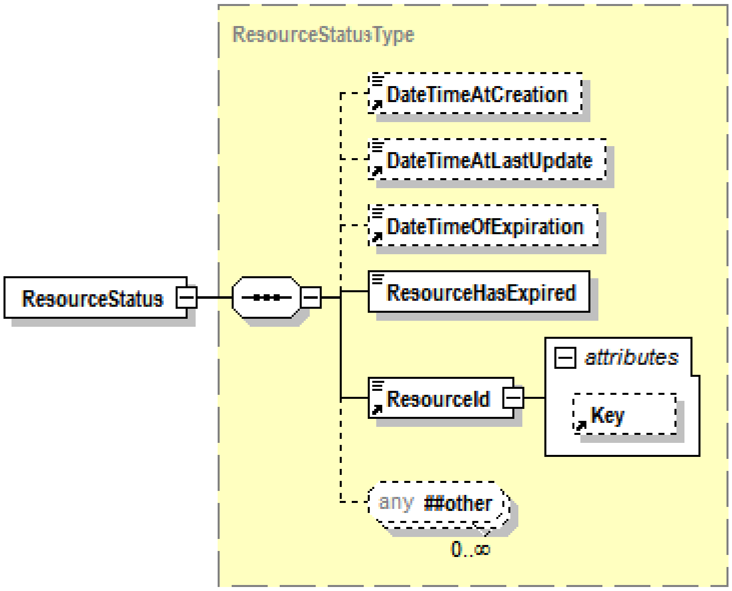


Figure 14 Resource Status Schema

7.6.1.2.1 DateTimeAtCreation

(dateTime) This element records the date and time when this resource was created by the Resource Service.

7.6.1.2.2 DateTimeAtLastUpdate

(dateTime) This element records the date and time when this resource was last updated by the Resource Service.

7.6.1.2.3 DateTimeOfExpiration

(dateTime) This element is settable only by an authorized administrator or through the RenewResource operation. It specifies the date and time this resource is to be expired.

At the time of Resource expiration, the Resource Service MUST choose either:

(1) delete the Resource, or
(2) Mark the Resource has expired by setting ResourceHasExpired to 'true' in the ResourceStatus element without deleting the Resource.

For security consideration, if (2) is chosen, the Resource Service MUST then:
(1) block all operations on the expired resource except for RenewResource operation.
(2) disallow any user (including administrator) to set any metadata element of the expired resource.

7.6.1.2.4 ResourceHasExpired

(Boolean) This REQUIRED element is automatically initialized to 'false' and set to 'true' by the Resource Service whenever the resource is expired, and the implementation choose not to delete the expired resource.

7.6.1.2.5 ResourceId

(int) This REQUIRED element uniquely identifies the Resource within the Service. The Service is responsible for implementing a Resource numbering scheme that will not allow two different Resources to coexist with the same ResourceId..

7.6.1.2.5.1 Key

(Boolean) The key is used to identify whether the ResourceId is the key of the Resource object.

7.6.1.2.6 List of ##other

(list of any) This list serves as extension points for the Resource Status properties while maintaining interoperability.

8 Theory of Operation

The Resource Service operates autonomously through three phases: initialization, online, and offline.

During MFD system start-up, the MFD System object may launch ResourceServiceStartup operation as part of the System's StartupService operation by which the Resource Service is created. On creation, the Resource Service is in Unknown state and enters its initialization phase during which all its service attributes and connected subunits are initialized. This phase may include test of its subunit(s) and self-testing of the Resource Service. After the initialization is successful, the Resource Service transits to the Down state which is a service offline state indicating that no user request is accepted, but the service will still respond to administrative requests. The service then continues the Startup operation from Down state to enter the Idle state and becomes ready for accepting service requests from Resource Clients.

The Resource Service accepts new service requests as long as it's in one of the two online states: Idle, and Processing. Receiving a resource transfer request in Idle state will generate an E.startRequest event. The service transits to Processing state, starts scheduling and processing the resource transfer such as put, get, replace, or delete a resource. While in the Processing state, any new resource transfer request will generate another E.startRequest event. When a resource transfer request is completed, an E.endRequest event is generated which causes the service to schedule and process the next available request. The service schedules and processes these requests in the Processing state until all requests are completed, at which time the last E.endRequest is generated which causes the service transits back to Idle state, or a shutdown operation may be performed and transits to 'Down' state if there is a C.shutdown condition pending.

While in an online (Idle or Processing) state, the service may receive E.critical events generated from critical errors of its subunit(s), service, or from the MFD system or other services. The service logs or reports the critical errors for impacted requests (resource transfer requests), sets C.critical condition and rejects all new resource transfer requests but continues servicing other requests in the same state. The service may also receive E.warning events generated from non-critical errors or E.endWarning from recovery of previous non-critical errors. When appropriate, an administrative ShutdownResourceService request can be manually sent that will transit the service to 'Down' state from either of the online states. From 'Down' state, implementation specific tests may be requested to diagnose and repair the faulted units or components to clear any critical condition. Then an administrative RestartResourceService request may be sent and the service transits from Testing or Down state to Idle state or Processing state. Note that an administrative RestartResourceService request or E.restart event received in any state initiates a sequence of state transitions. The service will transit to 'Down' state then proceed on to 'Idle'. Based on system conditions state transition on to Processing is possible.

While the service is online, performing an administrative DisableResourceService() operation will stop the Resource Service from accepting new resource transfer (e.g. storage or retrieval) requests, but still accepting other informational requests. An EnableResourceService() operation request will enable the service to accept new resource storage or retrieval requests again. While a StartupResourceService request can only be sent to the service for initial startup of the service, a RestartResourceService operation request received in Down, Idle, or Processing state will cause the same startup of the Resource Service and transits the service to Idle state and may proceed on to Processing state.

Before requesting a Resource Service, a user uses a local (via an MFD UI) or remote (via local network or Internet) Resource Client to to interact with a selected Resource Service. While the service is available, a Client application of an MFD Service can request one of the Resource Service operations specified in Section 8.1 that include DeleteResource, RetrieveResource, GetResourceElements, GetResourceServiceElements, ListResource, StoreResource, ReplaceResource, and SetResourceElement.

On StoreResource request, the Resource Service stores the specified Resource in a local or remote resource storage location. It is the implementation's responsibility to determine the target resource storage location for the

Resource Service. Once a resource is stored, a Resource Client can use the RetrieveResource request to retrieve the content of the resource identified by the ResourceId and the ResourceCreatorUserName.

A ListResource can be used to request a filtered list of Resources available to the requesting user, allowing the user to select a desired Resource. On a GetResourceElements request, the Resource Service obtains the user desired metadata of the Resource identified by the ResourceId and the ResourceCreatorUserName from the designated resource storage location. Similarly on a GetResourceServiceElements request, the Resource Service obtains the user desired metadata of the Resource Service currently selected and the ResourceCreatorUserName; the metadata returned is described in the user designated natural language. On a ReplaceResource request, the Resource Service replaces the existing resource identified by the ResourceId in resource storage location with the one supplied by the requesting user. On a SetResourceElements request, the Resource Service sets the elements (such as the DateTimeOfExpiration of the ResourceDescription of the resource identified by the ResourceId) to the desired content specified the user. On DeleteResource request, the Resource Service deletes the Resource in the resource storage location by ResourceId.

8.1 Basic Resource Service Operations

This section describes the semantics of all external interfaces that a PWG Resource Service MUST or is RECOMMENDED to support.

In general these interfaces are for use by a client application of the service. In the case when a MFD job service needs to directly interface with the Resource Service, it must implement the service client of these operations. Whether to implement a separate Resource Service Client for each MFD job service, or a common client component shared among multiple MFD job services is implementer's decision.

The Resource Service Client implemented by a MFD job service MAY choose to support only some of the REQUIRED Resource Service operations defined in this section. In all other cases, a conforming Resource Service Client MUST support all REQUIRED Resource Service operations defined herein.

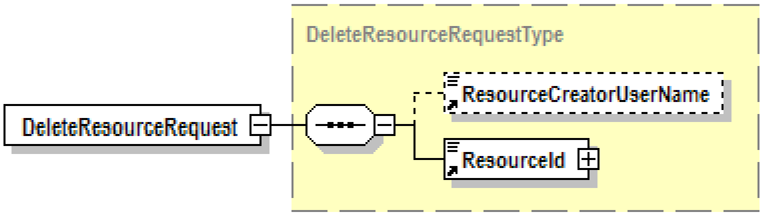
For security, a Resource Service MUST only allow authorized users to perform any of the basic operations. The operations and the level of access to resource categories for each operation authorized for a user is determined by the implementation (e.g. by the user's local site-policy). Every basic operation of the Resource Service described below includes ResourceCreatorUserName as an optional parameter in its request message. The ResourceCreatorUserNanme can be used to verify whether the requesting user is authorized for the operation, and his or her level of access to the resources through the operation.

8.1.1 DeleteResource

This REQUIRED operation permanently removes a Resource from its resource storage location by ResourceId.

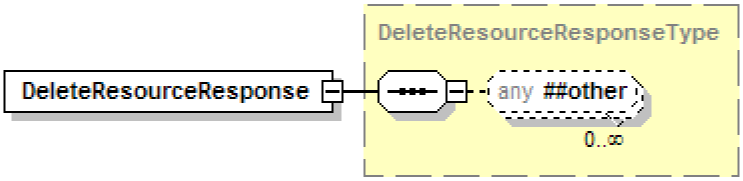
8.1.1.1 DeleteResourceRequest

Below is the DeleteResource request message schema. The request contains an OPTIONAL element which is the ResourceCreatorUserName and a REQUIRED element which is the ResourceId. The key of the ResourceId is used to identify that this element is the key of the Resource object.



8.1.1.2 DeleteResourceResponse

The response MUST be a successful return or a fault. The response may contain vendor extension parameters.

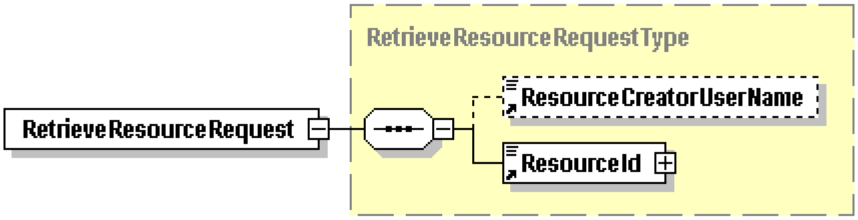


8.1.2 RetrieveResource

This REQUIRED operation allows a Resource Client to retrieve the content of a Resource.

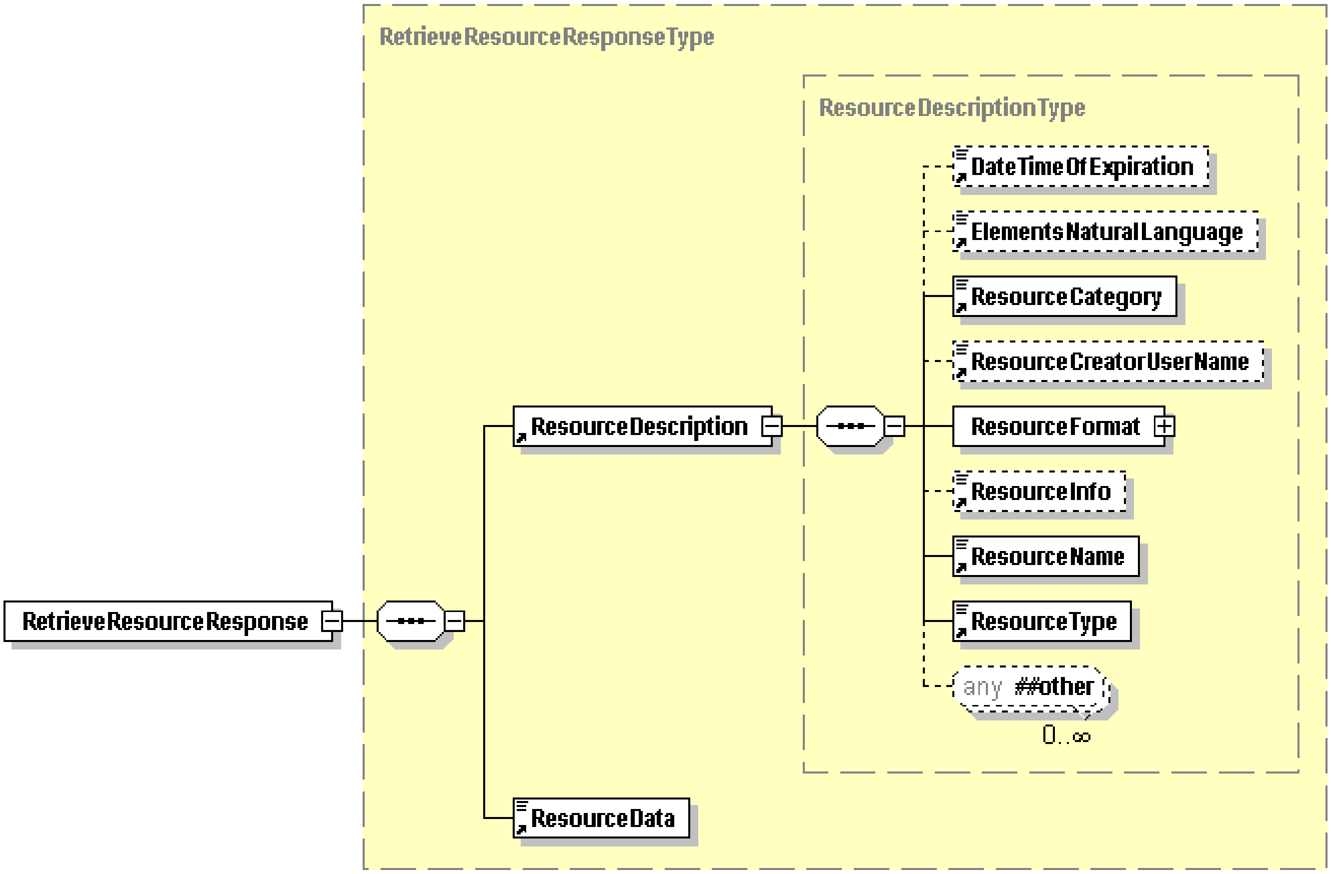
8.1.2.1 RetrieveResourceRequest

This request retrieves the content of a Resource identified based on the REQUIRED ResourceId and the OPTIONAL ResourceCreatorUserName. The service will use the most trustworthy layer of the protocol to arrive the identity of the Requesting User ID.



8.1.2.2 RetrieveResourceResponse

The response of RetrieveResource request returns the ResourceDescription elements first, followed by the content of the resource. A Resource may contain binary data which must be transported in a binary-safe mechanism (e.g. MTOM [MTOM]).

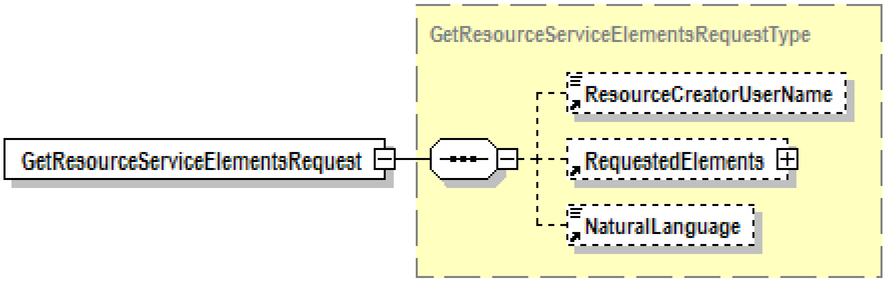


8.1.3 GetResourceServiceElements

This REQUIRED operation allows a Resource Client to retrieve the metadata of a Resource Service.

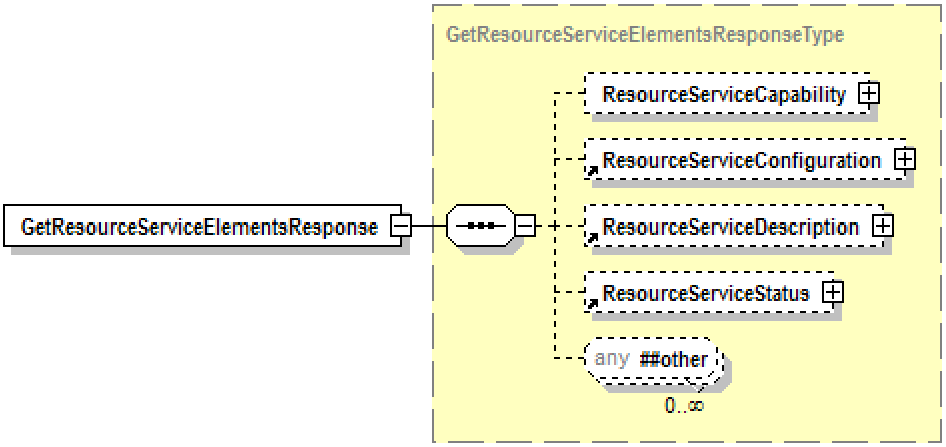
8.1.3.1 GetResourceServiceElementsRequest

This request gets a list of summarized information that only contains top level information of a list of requested elements of the Resource Service targeted by the resource client. The ResourceCreatorUserName can be used to authenticate the requesting user for accessing these service elements for security. NaturalLanguage is used to specify the natural language used for the returned description text of the service elements.


GetResourceServiceElementsRequestType
ResourceCreatorUserName
GetResourceServiceElementsRequest
RequestedElements
NaturalLanguage

8.1.3.2 GetResourceServiceElementsResponse

The response of GetResourceServiceElements request should include one or more of the top level Resource Service elements requested: ResourceServiceCapability, ResourceServiceConfiguration, ResourceServiceDescription, ResourceServiceStatus, and other vendor specific extension.

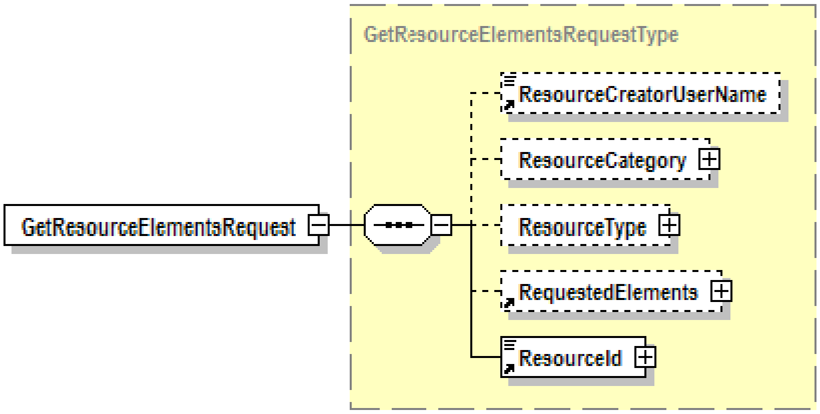

GetResourceServiceElementsResponseType
GetResourceServiceElementsResponse
ResourceServiceCapability
ResourceServiceConfiguration
ResourceServiceDescription
ResourceServiceStatus
any ##other
0..∞

8.1.4 GetResourceElements

This REQUIRED operation allows a Resource Client to retrieve the metadata of a Resource.

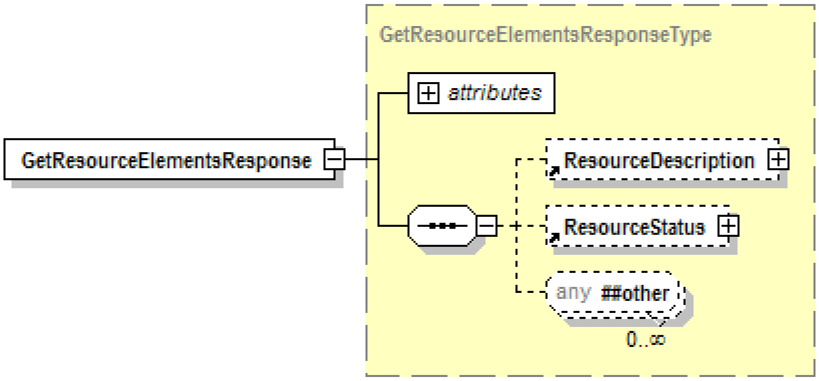
8.1.4.1 GetResourceElementsRequest

This request gets a list of summarized information for a list of requested elements of the Resource identified by the ResourceId and the ResourceCreatorUserName. The ResourceCreatorUserName can be used to determine the access rights of the requesting user to these resource elements for security. This request gets a list of summarized information that only contains top level elements of ResourceDescription and ResourceStatus of Resources, filtered by three optional elements: ResourceCreatorUserName, ResourceCategory, and ResourceType.



8.1.4.2 GetResourceElementsResponse

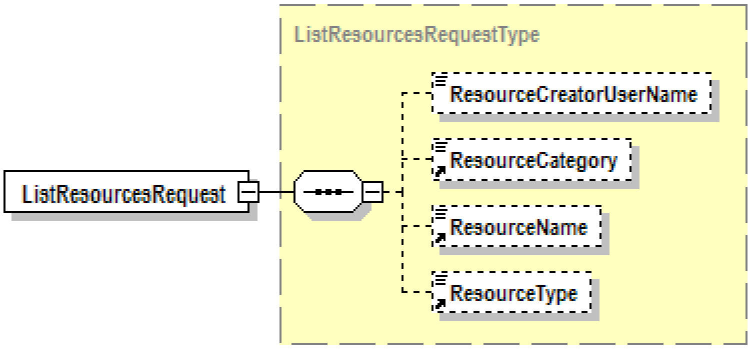
The response message from GetResourceElements request MUST contain only top level elements of ResourceDescription and ResourceStatus. The ResourceDescritpion element contains REQUIRED ResourceType, ResourceCategory, ResourceName, and ResourceFormat. ResourceType can be used by Resource Client to interpret the retrieved resource data.



8.1.5 ListResources

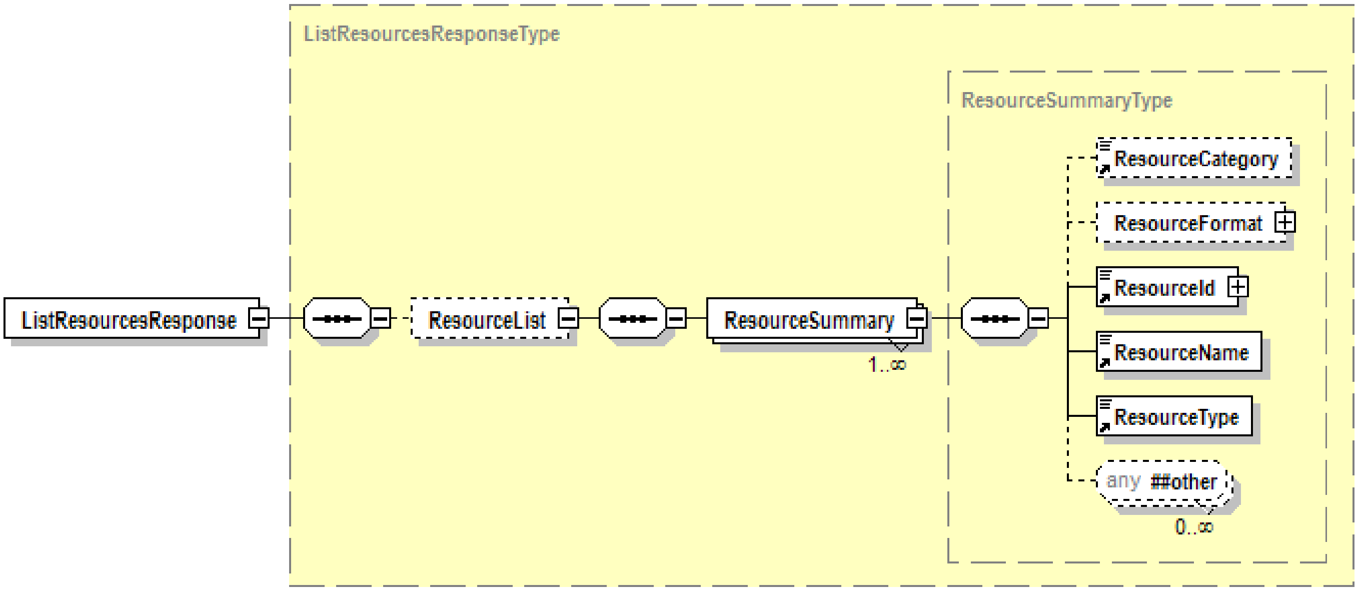
This REQUIRED operation allows a Resource Client to get a list of Resource. The list of Resources returned will provide some summary information on each Resource. The Resource Service SHOULD restrict the list of Resources returned based on ResourceCreatorUserName, ResourceCategory, ResourceType, ResourceName. The service will use the most trustworthy layer of the protocol to arrive the identity of the Requesting User ID.

8.1.5.1 ListResourcesRequest



8.1.5.2 ListResourcesResponse

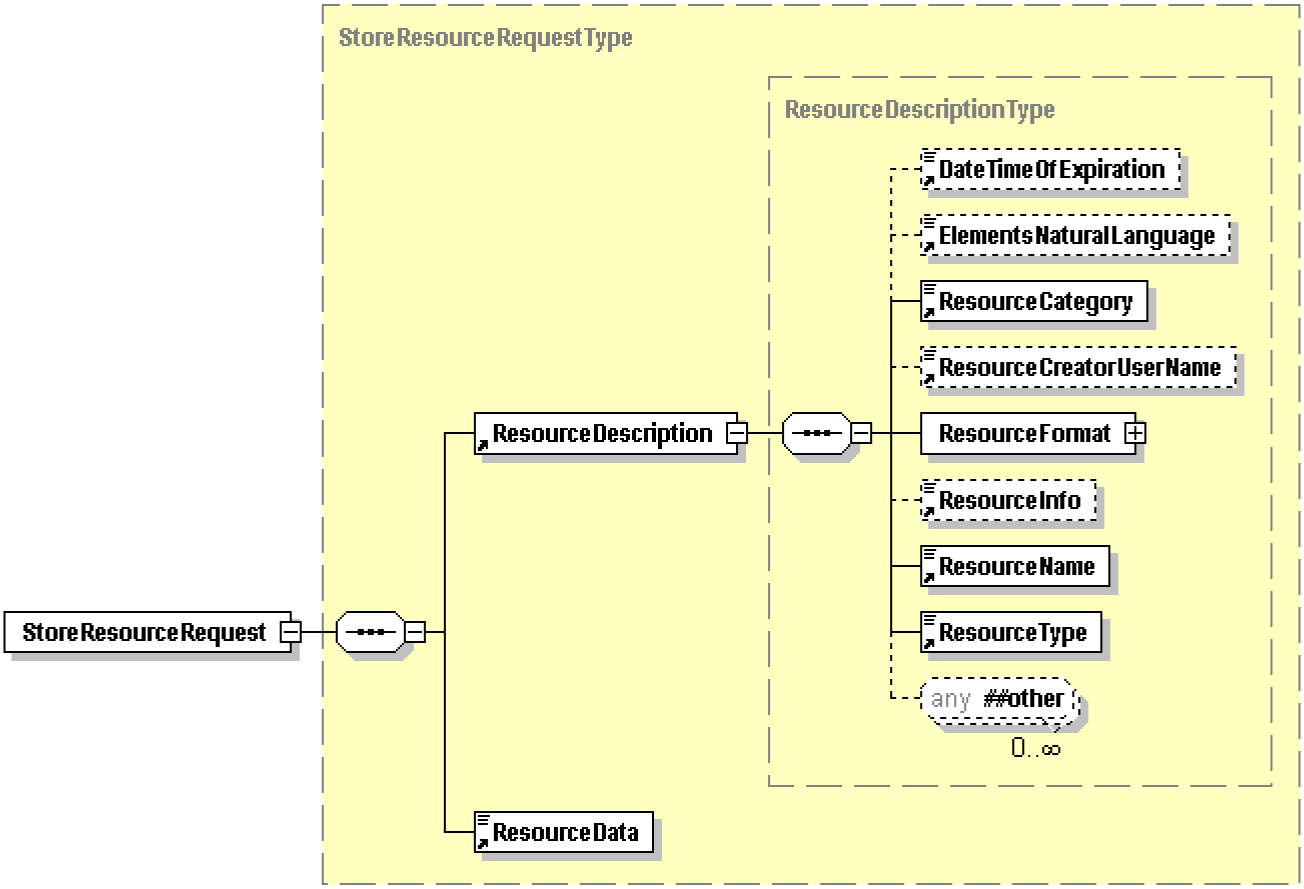
The response to the ListResource request only returns the top level summary information about the resources matching the requested ResourceType, ResourceCategory, and ResourceCreatorUserName. The ResourceSummary for each returned resource only contains ResourceCategory, ResourceFormat, ResourceId, ResourceName and ResourceType elements.



8.1.6 StoreResource

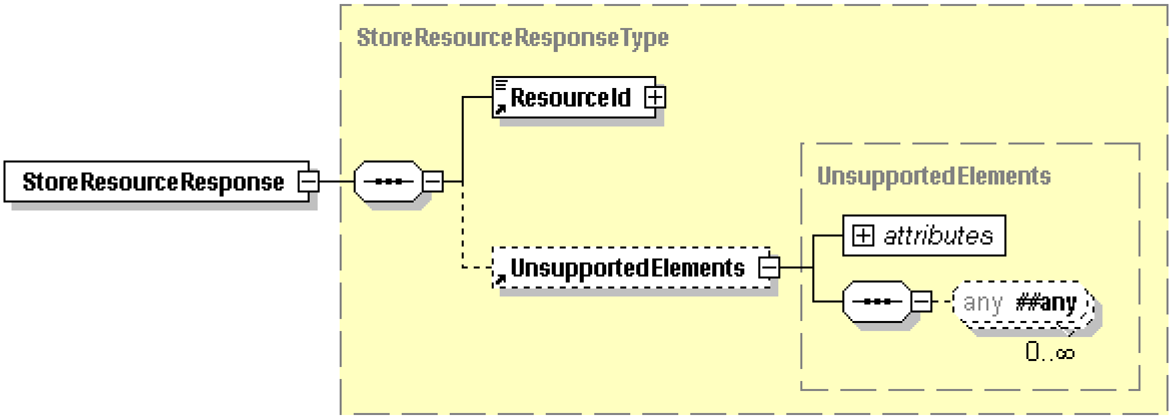
This REQUIRED operation stores a new Resource with the specified Resource descriptions at the resource storage location connected to this Resource Service instance. This operation MUST generate a GUID for the Resource and validate that the requested Resource is well-formed before the Resource is stored at the storage location if the resource is a Template. A Resource may contain binary data which must be transported in a binary-safe mechanism (e.g. MTOM [MTOM]).

8.1.6.1 StoreResourceRequest



8.1.6.2 StoreResourceResponse

The response MUST contain the ResourceId to identify the Resource in the StoreResource request for either a successful return or a fault. The response below indicates that vendors may extend the response with additional information. If the resource being stored has elements or element values that are unsupported by the service, a success status with the UnsupportedElements MUST be returned by the service.

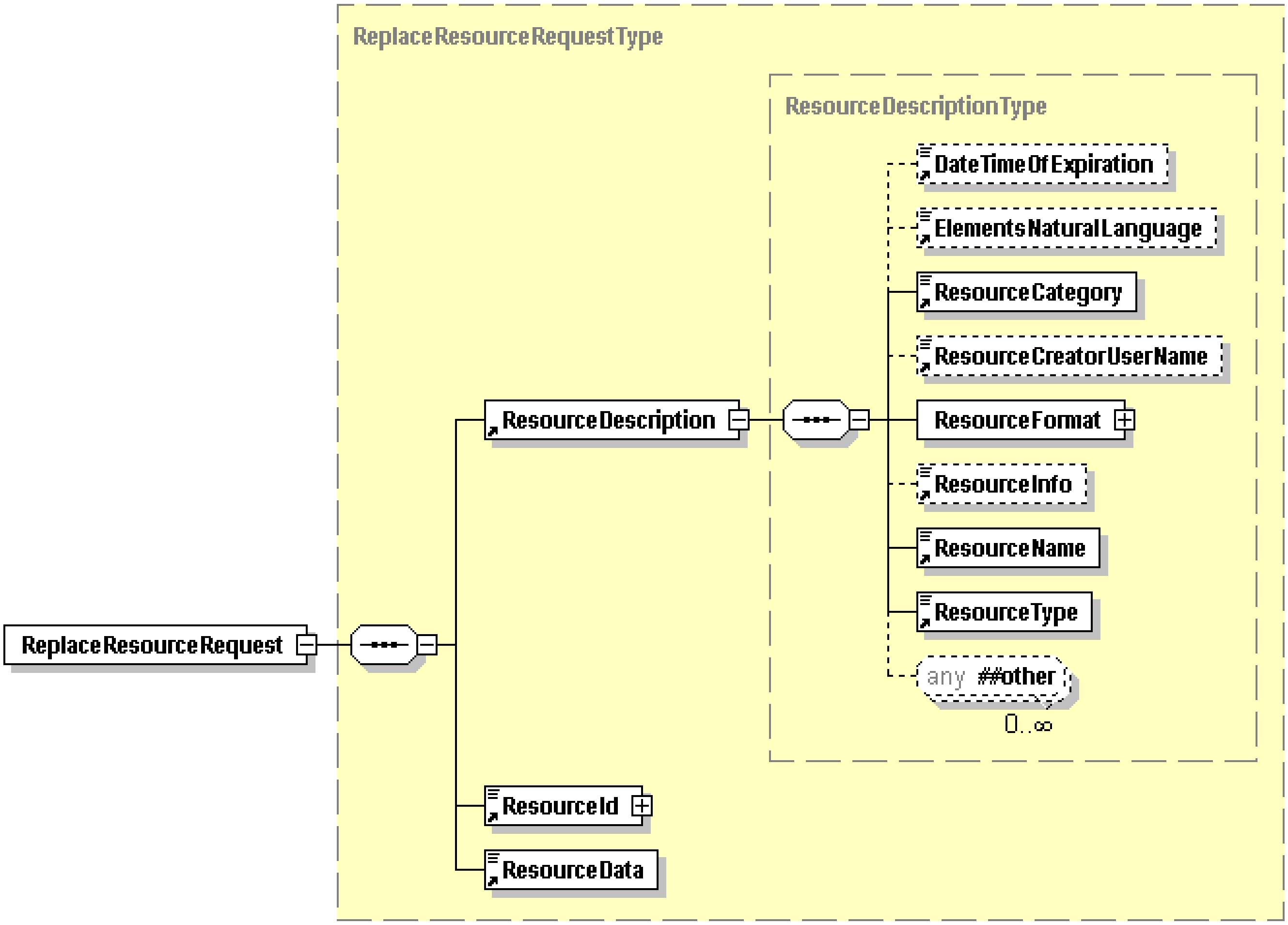


8.1.7 ReplaceResource

This OPTIONAL operation replaces the ResourceEntry that has the matching ResourceId at the connected.resource storage location.

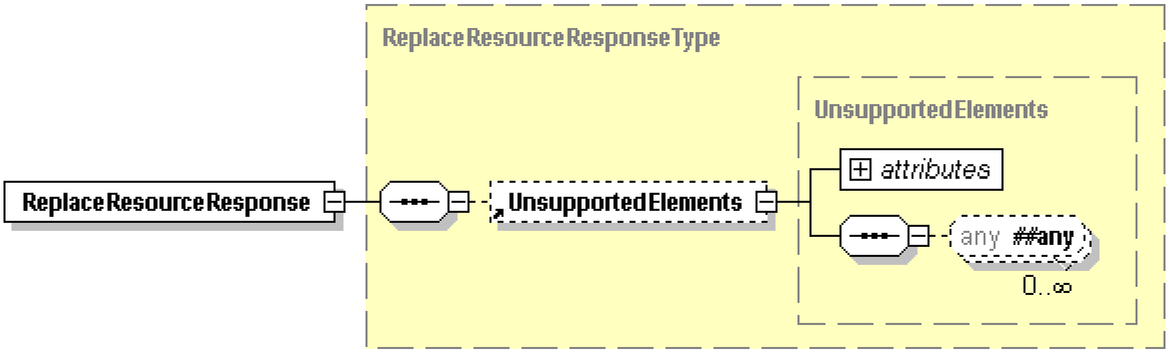
8.1.7.1 ReplaceResourceRequest

This request replaces the metadata and the content of the resource that matches the resource with the ResourceId stored at the resource storage location, The Resource Client MUST supply at least three elements in the ResourceDescription which are ResourceType ResourceFormat, and ResourceCategory. The service MUST verify that the replacing resource has the same ResourceType, ResourceFormat, and ResourceCategory as the one currently stored at the storage location. For maintaining consistency, the service MUST never update the ResourceCreatorUserName, the ResourceFormat, ResourceCategory, and the ResourceType of the ResourceDescription of the existing resource. A Resource may contain binary data which must be transported in a binary-safe mechanism (e.g. MTOM [MTOM]).



8.1.7.2 ReplaceResourceResponse

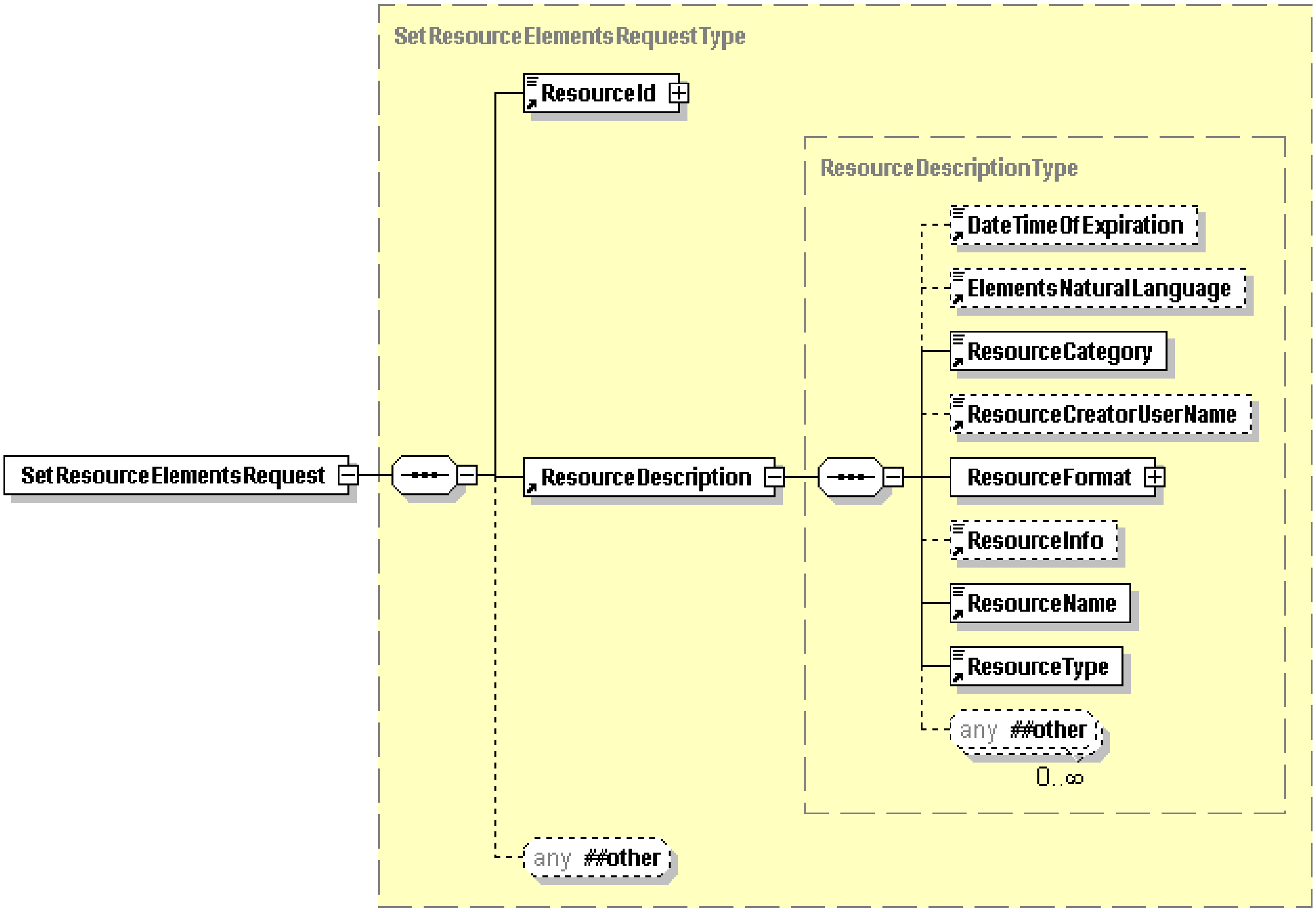
The response MUST be a successful return or a fault. The response below indicates that vendors may extend the response with additional information. If the resource being replaced has descriptive elements or element values that are unsupported by the service, a success status with the UnsupportedElements MUST be returned by the service.



8.1.8 SetResourceElements

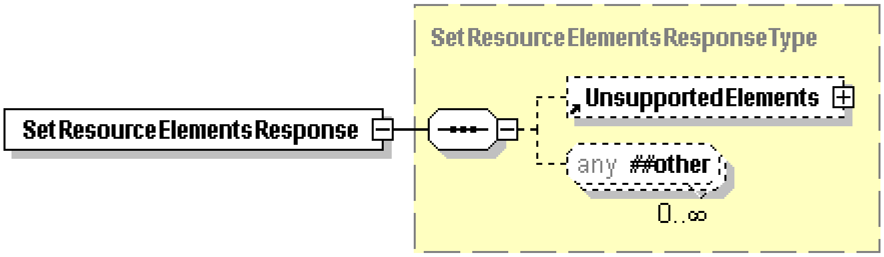
This OPTIONAL request updates the metadata of the Resource that has the matching ResourceId indicated by the input parameter ResourceId to the ResourceDescription passed in as a request parameter. The Resource Client MUST supply at least three elements in the ResourceDescription which are ResourceType, ResourceFormat, and ResourceCategory. The service MUST verify that the replacing resource has the same ResourceType, ResourceFormat, and ResourceCategory as the one currently stored at the storage location. For maintaining consistency, the service MUST never update the ResourceCreatorUserName, the ResourceFormat, the ResourceCategory, and the ResourceType of the ResourceDescription of the existing resource.

8.1.8.1 SetResourceElementsRequest



8.1.8.2 SetResourceElementsResponse

The response MUST be a successful return or a fault. The response may contain vendor extension parameters. If the resource metadata being set has elements or element values that are unsupported by the service, a success status with the UnsupportedElements MUST be returned by the service.



8.2 Administrative Resource Service Operations

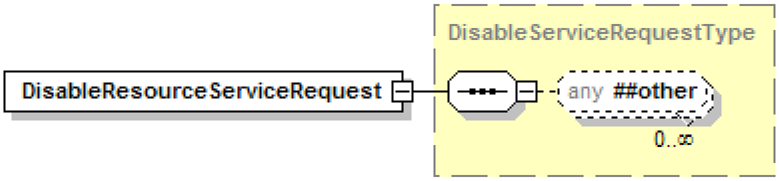
All administration operations are OPTIONAL for Resource Service to support. Administrative operations are limited to the authorized administrators only. Administrative operations available for Resource Service are disable, enable, shutdown, and startup. The request and response messages of these operations are the same for all MFD services. These messages are generically defined in PWG Administrative Operations and reused in the message body of the request and response messages of the corresponding Resource Service operations.

In all the administrative operations described below, each request or response of an operation may contain vendor extended parameters.

8.2.1 DisableResourceService

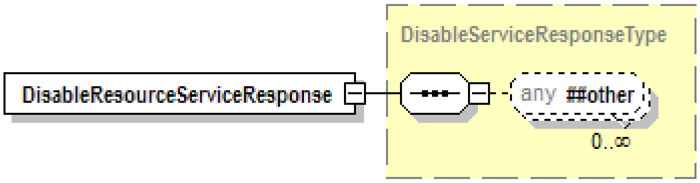
This OPTIONAL operation prevents the Resource Service from accepting any new resource retrieval or storage operation request (e.g. RetrieveResource, StoreResource, ReplaceResource, SetResourceElements operation). The Resource Service is still able to process ListResources, GetResourceElements, GetResourceServiceElements operations.

8.2.1.1 DisableResourceServiceRequest



8.2.1.2 DisableResourceServiceResponse

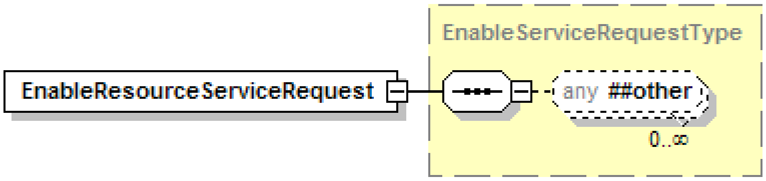
The response MUST be a successful return or a fault. The response may contain vendor extension parameters.



8.2.2 EnableResourceService

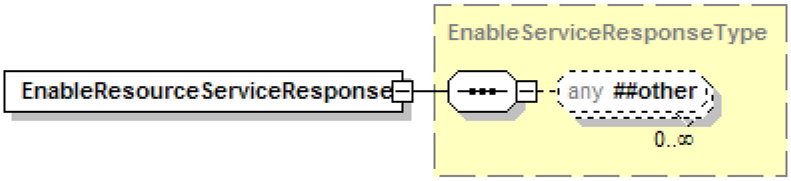
This OPTIONAL operation allows a previously disabled Resource Service to accept new resource retrieval or storage operation requests (e.g. StoreResource operation).

8.2.2.1 EnableResourceServiceRequest



8.2.2.2 EnableResourceServiceResponse

The response MUST be a successful return or a fault. The response may contain vendor extension parameters.

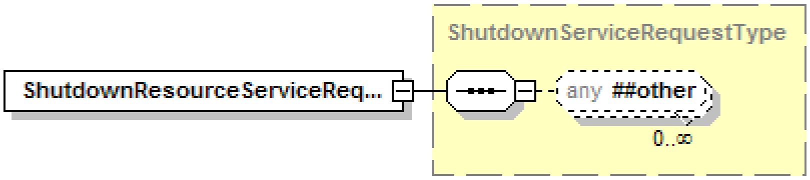


8.2.3 ShutdownResourceService

This OPTIONAL administrative operation causes the Resource Service to stop processing requests, complete the processing of any currently processing jobs as soon as the implementation allows, stop accepting new requests, and causes the Resource Service to terminate in an orderly fashion.

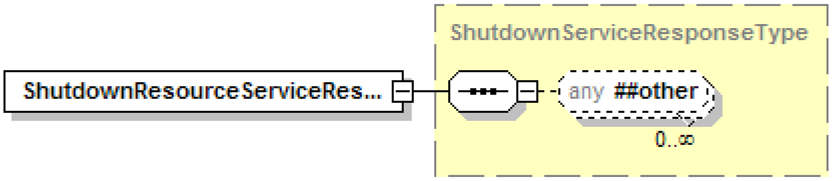
8.2.3.1 ShutdownResourceServiceRequest

This request stops all ResourceService operations, except for StartupResourceService.



8.2.3.2 ShutdownResourceServiceResponse

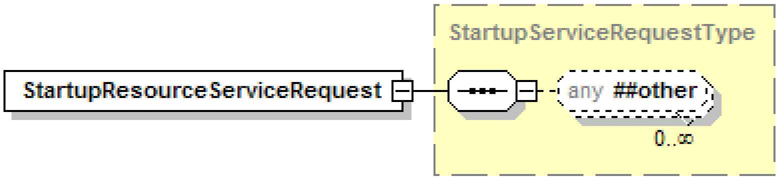
The response MUST be a successful return or a fault. The response may contain vendor extension parameters.



8.2.4 StartupResourceService

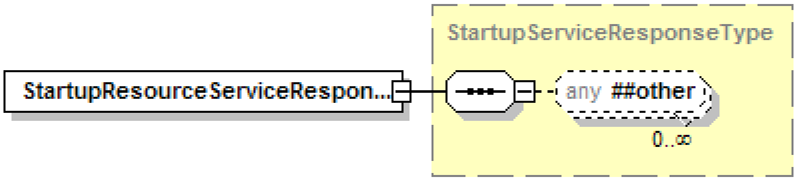
This OPTIONAL administrative operation causes a new instance of a Resource Service to begin initialization and then move from Down state to Idle state if possible (i.e. no hardware or software errors occur that would prevent normal operation).

8.2.4.1 StartupResourceServiceRequest



8.2.4.2 StartupResourceServiceResponse

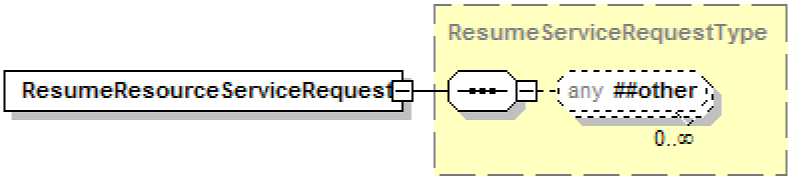
The response MUST be a successful return or a fault. The response may contain vendor extension parameters.



8.2.5 RestartResourceService

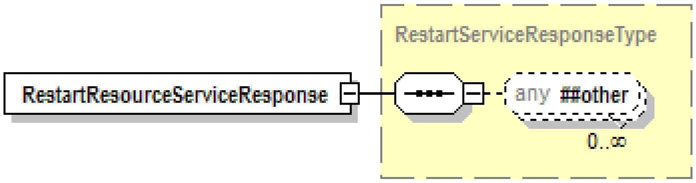
This OPTIONAL administrative operation causes a previously shut down instance of a Resource Service to begin initialization and then move to Idle state if possible (i.e. no hardware or software errors occur that would prevent normal operation). The same operation can be requested at any other service state that will cause the same service initialization and transits the service to Idle state.

8.2.5.1 RestartResourceServiceRequest



8.2.5.2 RestartResourceServiceResponse

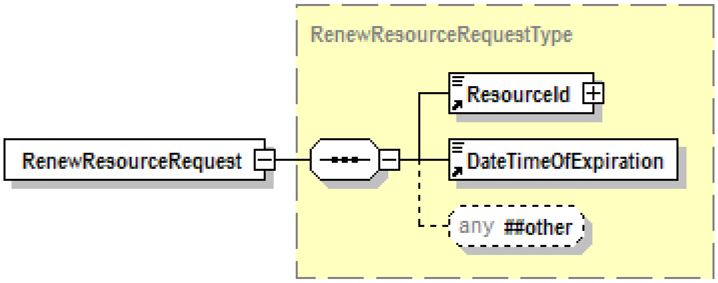
The response MUST be a successful return or a fault. The response may contain vendor extension parameters.



8.2.6 RenewResource

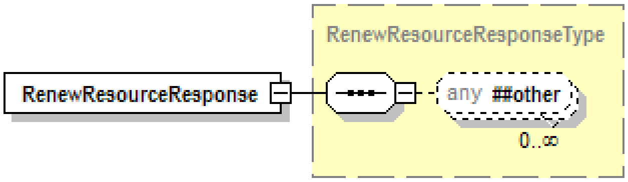
This OPTIONAL administrative operation sets the new DateTimeOfExpiration in the ResourceDescription and the ResourceStatus element of the Resource identified by its ResourceId.

8.2.6.1 RenewResourceRequest



8.2.6.2 RenewResourceResponse

The response MUST be a successful return or a fault. The response may contain vendor extension parameters.



9 Conformance Requirements

This section describes conformance issues and requirements. This document introduces model entities such as objects, operations, elements, element syntaxes, and element values. These conformance sections describe the conformance requirements which apply to these model entities.

9.1 Client Conformance Requirements

A conforming client MUST support all REQUIRED operations as defined in this document. For each parameter included in an operation request, a conforming client MUST supply a value whose type and value syntax conforms to the requirements of the Model document as specified in Section 8. A conforming client MAY supply any extensions in an operation request, as long as they meet the requirements in Section 9.4.

When sending a request, a conforming client NEED NOT supply any parameters that are indicated as OPTIONALLY supplied by the client.

A client MUST be able to accept any of the elements defined in the model, including their full range that may be returned to it in a response from a Resource Service

An operation response may contain elements and/or values that the client does not expect. Therefore, a client implementation MUST gracefully handle such responses and not refuse to interoperate with a conforming Resource Service that is returning extended elements and/or values that conform to Section 9.4. Clients may choose to ignore any parameters, elements, or values that they do not understand.

9.2 Resource Service Conformance Requirements

This section specifies the conformance requirements for conforming implementations with respect to objects, operations, and attributes.

9.2.1 Objects

Conforming implementations MUST implement all of the model objects as defined in this specification in the indicated sections:

Section 7 - Resource Service Model

Conforming Resource Service MUST support all standard types, categories, or format of resource object except as restricted by the site policy.

9.2.2 Operations

Conforming Resource Service implementations MUST implement all of the REQUIRED model operations, including REQUIRED responses, as defined in this specification in the indicated sections:

DeleteResource (section 8.1.1) REQUIRED
RetrieveResource (section 8.1.2) REQUIRED
GetResourceElements (section 8.1.4) REQUIRED
GetResourceServiceElements (section 8.1.3) REQUIRED
ListResources (section 8.1.5) REQUIRED
StoreResources (section 8.1.6) REQUIRED

Conforming Resource Service MUST support all REQUIRED operation elements and all values of such elements if so indicated in the description. Conforming Resource Service MUST ignore all unsupported or unknown operation elements received in a request.

9.3 Resource Service Elements

Conforming Resource Service MUST support all of the REQUIRED object elements, as defined in this specification.

If an object supports an element, it MUST support only those values specified in this document or through the extension mechanism described in section 9.4. It MAY support any non-empty subset of these values. That is, it MUST support at least one of the specified values and at most all of them.

9.4 Extensions

Conforming Resource Service MAY support extensions. To extend the model the extensions MUST be fully qualified. The qualified name MUST NOT be in the PWG target namespace. When extending the model with new elements the new elements MUST be added at the extension points at the end of the associated sequence of elements. Extended values for elements MUST conform to the extension patterns defined in the element schema. Implementers are free to add vendor specific operations to the service.

10 PWG and IANA Registration Considerations

Once the specification and associated schema is published it will require a new version of the specification to register extensions to the Resource Service model. Vendors may use extensions in their own namespace until such time as an update to the specification is under review. At that time the extension can be registered with the PWG and included in the PWG specification.

11 Internationalization Considerations

The values of element that are part of enumeration are represented by keywords. Keywords are never localized by the device. The client may convert the values into a form acceptable to the client. This includes not only localization but also transformations into graphical representation. The elements with an extensible set of keywords are represented by the union of an enumeration of keywords and a pattern for new values.

Some of the elements have values that are Resource Service generated strings (e.g. StateMessages). In each operation request, the client identifies a natural language that affects the Resource Service generated strings returned by the Resource Service in operation responses. The Resource Service MUST provide the localized value as requested by the user for any supported natural languages [RFC 2277]. A request for a language not supported results in a response with the string in the default localization.
The final category of string values are those supplied by administrator or user (e.g. ResourceName). No localization is performed on these strings and they are returned in operation responses as set by the administrator or user.

12 Security Considerations

12.1 Protection of User's Resources

When the confidentiality or integrity or both of a user's resource need to be protected, a site-specific access control policy may be defined, for example based on the ResourceCreatorUserName for which the access rights associated with the user can be verified for operations accessing the resources. Hence all basic resource service operations MUST be limited to authorized users for security considerations; all administrative resource service operations are restricted to authorized administrators only.

For protection of the resources transmitted over the network between a Resource Service and a Resource Client, it is RECOMMENDED that at a minimum the Resource Service should support one or more secure communication protocol required by the local site policy.

A site policy administrator SHOULD be responsible to manage the site security policy to ensure consistency with the site security requirements.

12.2 Security Threats from Executable Resources

Resources with a ResourceCategory of 'Executable' MUST be handled with special care by implementations of the Resource Service. Such resources can pose serious threats to the integrity of the Imaging System that hosts the Resource Service. In particular, such Resources can be used to introduce Trojan Horses to the Imaging System. If an implementation of the Resource Service supports Executable resources, then that implementation MUST restrict the storage of such resources (e.g., to authorized administrators and manufacturers) and SHOULD verify the safety of such resources.

12.3 Security Threats from Static Resources

Resources with ResourceCategory of 'Static' SHOULD be treated with special care by implementations of the Resource Service. Fonts that have associated Intellectual Property rights (e.g., as part of their network licenses) can pose serious threats to the availability of the Imaging System that hosts the Resource Service - security audits can result in the shutdown or physical removal of the Imaging System. If an implementation of the Resource Service supports Static resources that have associated Intellectual Property rights, then that implementation SHOULD restrict the storage of such resources (e.g., to authorized administrators and manufacturers) and SHOULD restrict the retrieval of such resources (e.g., to a configured group of authorized users).

12.4 Security Threats to Metadata of All Resources

The metadata of a resource may contain elements associated with a user's access rights to the resource such as ResourceCreatorUserName, or validity of the resource such as DateTimeOfExpiration. Hence the Resource Service operations that provide access to these metadata of all resources SHOULD be restricted to authorized users. The default access policy of the metadata SHOULD restrict modification to creator, owner, or authorized administrators. Some metadata such as DateTimeOfExpiration or manufacturer's copyright should be READONLY once having been created by the authorized role (e.g. manufacturer), even owner or administrator may not be allowed to modify.

When a resource license is expired, if the Resource Service chooses not to delete the resource, only mark the resource has expired, the Resource Service MUST then:
(1) block all operations on the expired resource except for RenewResource.
(2) disallow any user (including administrator) to set any metadata element of the expired resource.

13 References

13.1 Normative References

[PWG5100.5]
PWG 5100.5-2003, "Internet Printing Protocol (IPP): Document Object", October 31, 2003, D. Carney, T. Hastings, and P. Zehler, ftp://ftp.pwg.org/pub/pwg/candidates/cs-ippdocobject10-20031031-5100.5.pd

[PWG5106.1-2007]
PWG 5106.1-2007, "The Printer Working Group (PWG) Standardized Imaging System Counters 1.1", April 27, 2007, H. Lewis, I. McDonald, J. Thrasher, W. Wagner, and P. Zehler, ftp://ftp.pwg.org/pub/pwg/candidates/cs-wimscount11-20070427-5106.1.pdf

[PWG5108.02-2009]
PWG5108.02-2009, "Network Scan Service Semantic Model and Service Interface 1.0", April 10, 2009, Nancy Chen and Peter Zehler. ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm20-scan-20090410-5108.02.pdf

[RFC2119]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC 2277]

H. Alvestrand, "IETF Policy on Character Sets and Languages", RFC 2277, January 1998.

[RFC2426]

F. Dawson, T. Hawes, "vCard MIME Directory Profile", RFC 2426, September 1998.

[RFC 2790]
S. Waldbusser, P. Grillo, "Host Resources MIB", RFC 2790, March 2000.

[RFC2911]
T. Hastings, R. Herriot, R. deBry, S. Isaacson, P. Powell, "Internet Printing Protocol/1.1 Model and Semantics", September 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2911.txt

[RFC4646]
H. Alvestrand, "Tags for the Identification of Languages", January 2001, http://www.ietf.org/rfc/rfc4646.txt, supersede RFC 3066,

[RFC 3805]
R. Bergman, H. Lewis, I. McDonald, "Printer MIB v2", RFC 3805, June 2004.

[MTOM]
Martin Gudgin, Noah Mendelsohn, Mark Nottingham, Herve Ruellan, "SOAP Message Transmission Optimization Mechanism", W3C Recommendation, 25 January 2005. http://www.w3.org/TR/soap12-mtom/

13.2 Informative References

[RFC5013]
J. Kunze, T. Baker, “The Dublin Core Metadata Element Set”, RFC 5013, August 2007

[Others TBD]

14 Author’s Address

Nancy Chen
Solutions and Technology
Oki Data
2000 Bishops Gate Blvd.
Mt. Laurel, NJ 08003
Phone: 856 222 7006
Fax: 856 222 5130
Email: nchen@okidata.com

Peter Zehler
Xerox Research Center Webster
Email: Peter.Zehler@Xerox.com
Voice: (585) 265-8755
FAX: (585) 265-7441
US Mail: Peter Zehler
Xerox Corp.
800 Phillips Rd.
M/S 128-25E
Webster NY, 14580-9701

Ira McDonald
High North Inc.
579 Park Place Saline, MI 48176
Phone: 734-944-0094
Email: blueroofmusic@gmail.com

Additional contributors: (Still to be updated)

Shah Bhatti	
Lee Farrell -	Canon
Mike Fenelon -	Microsoft
Andrey Savov -	Toshiba
Harry Lewis -	IBM
Ira McDonald -	High North, Inc.
Joe Murdock -	Sharp
Glen Petrie -	Epson
Jerry Thrasher -	Lexmark
Bill Wagner -	TIC
David Whitehead -	Lexmark
Craig Whittle -	Sharp