Resource Service State Transitions and Theory of Operation

1.1.1 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.

1.1.1.1 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 for the service state diagram.

1.1.1.2 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)

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

(2) 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)

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

(4) Testing

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

1.1.1.2.1 Service State Transition By Operations

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

1.1.1.2.2 Service State Transition By Events

SERVICE STATE MACHINE (Events)				
Input	**State**			
	Down	**Testing**	**Idle**	**Processing**
Event (Condition)	**Action (new state)**	**Action (new state)**	**Action (new state)**	**Action (new state)**
E.critical	N/A	critical	critical	critical
E.endCrit	N/A	endCritical	error	error
E.endRequest (C.shutdown)	N/A	error	error	shutdown (Down)
E.startRequest (C.disabled)	N/A	error	error	ignore
E.endTest	N/A	testReport (Down)	error	error
E.endWarn	N/A	endWarning	endWarning	endWarning
E.Startup (Note 2)	restart (Idle)	error	error	error
E.warning	N/A	warning	warning	warning

1.1.1.2.3 Detailed Service State Transition Diagram

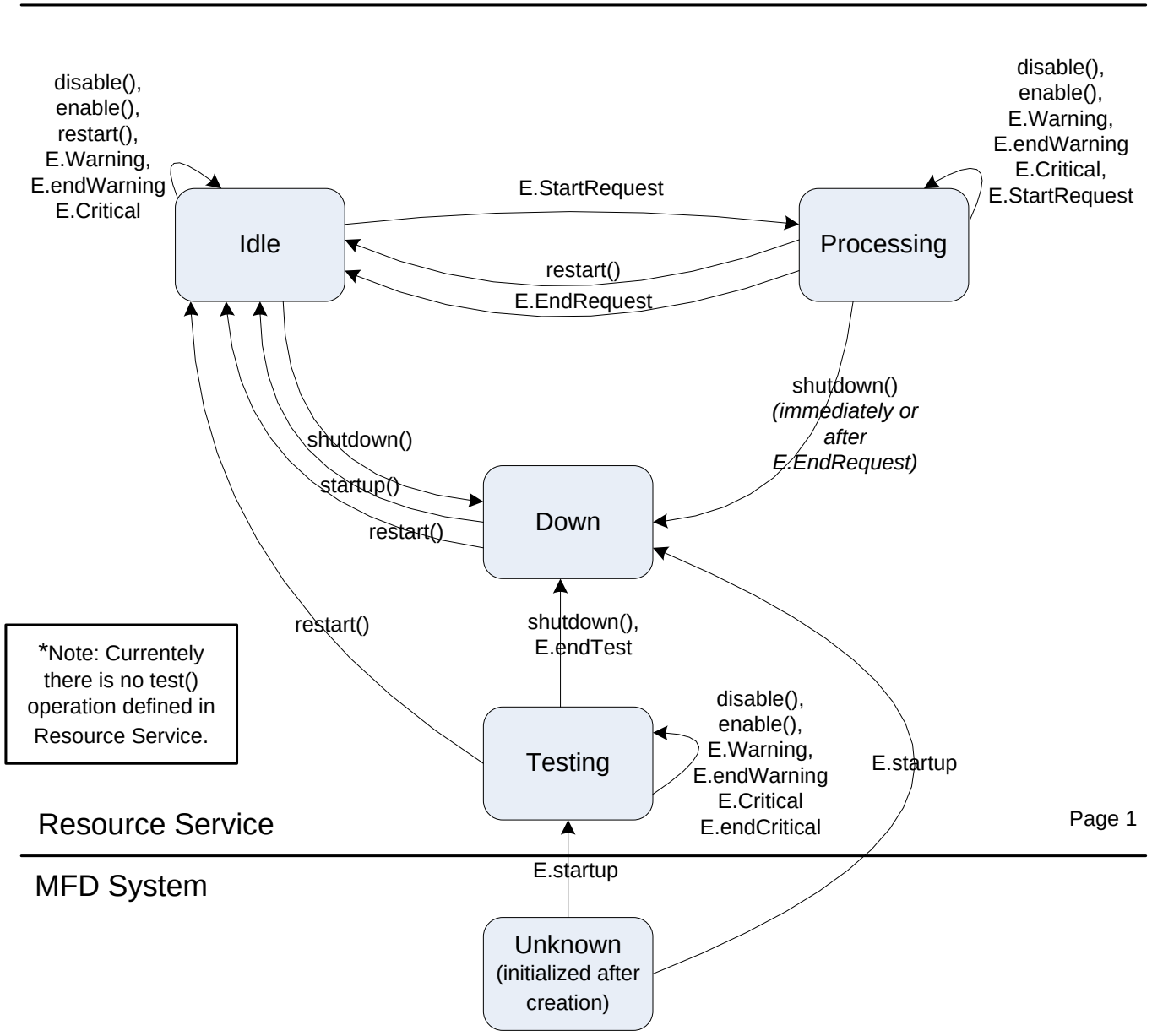

Figure 1 Detailed Service Transition Diagram

2 Theory of Operation

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

During MFD system start-up, the Resource Service is created. On creation, the Resource Service enters its initialization phase during which all its service attributes and connected subunits are initialized. This phase may include test of the Storage 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 requests are accepted. The MFD system then sends an E.startup event with which the Resource Service performs a startup which brings the service online after authenticated and registered its service with a service directory or announced its service to the network domain in which it resides. The Resource Service then enters the "Idle" state and becomes ready for service discovery and 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 new service request in Idle state will generate an E.startRequest event and the service transit to Processing state and start processing the request. While in the Processing state, receiving a new request will generate another E.startRequest event. The service processes these requests in the Processing state until all requests are completed, then an E.EndRequest will be generated which transits the service back to Idle state.

While online, the service may receive E.critical events generated from critical errors of its own subunits or service or sent from the MFD system, or other services. The service may also receive E.warning or E.endWarn events generated from non-critical errors or recovery from previous non-critical errors. Performing an administrative DisableResourceService() operation while the service is online will stop the Resource Service from accepting new resource storage or retrieval requests, but still continue to accept other informational requests. An EnableResourceService() operation request while the Service is disabled will enable new resource storage or retrieval requests to be accepted again. When necessary, the Resource Service can be manually shutdown by an authorized administrator by sending a ShutdownResourceService request, and later manually restarted by sending a RestartResourceService request.

Before requesting a Resource Service, a user uses a local (via an MFD UI) or remote (via local network or Internet) Resource Client to discover and select the desired target Resource Service. While the service is available, a Client application of an MFD Service can request one of the Resource Service operations specified in Sections **Error! Reference source not found.** that include DeleteResource, GetResource, GetResourceElements, GetResourceServiceElements, ListResource, PutResource, ReplaceResource, and SetResourceElement.

On PutResource request, the Resource Service stores the specified Resource in a local or remote Resource Repository. It is implementation's responsibility to determine the target Resource Repository for the Resource Service. Once a resource is stored, a Resource Client can use the GetResource 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 Repository. Similarly on a GetResourceServiceElements request, the Resource Service obtains the user desired metadata of the Resource Service identified by the Resource Service Id 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 repository 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 end user. On DeleteResource request, the Resource Service deletes the Resource in the repository by ResourceId.