



# TS-003

## Tool Resource

---

**Abstract:** This extension to NTAF Tool Automation harness describes a framework for specifying resource data.

**Authors:** Brian Bonnett, Mickael Graham, Kenneth Green, Elemér Lelik, Eric Miller, Kris Raney

**Copyright:** © 2013, Network Test Automation Forum. All rights reserved.

**Status:** Final

**Revision:** 01

**Revision date** August 2013

**Submission:** ntaf ts-003

## Legal Notices

This Specification has been created by the Network Test Automation Forum (NTAF). NTAF reserves the rights to at any time add to, amend, modify or withdraw all or any portion of this Specification.

Note: All parties who in any way intend to use this Specification for any purpose, are hereby put on notice that the possibility exists that practicing under this Specification may require the use of inventions covered by the patent rights held by third parties. By publication of this Specification NTAF makes no representation or warranty whatsoever, whether expressed or implied, that practicing under this will not infringe any third party rights, nor does NTAF make any representation or warranty whatsoever, whether expressed or implied, with respect to (1) any claim that has been or may be asserted by any third party, (2) the validity of any patent rights related to any such claim, (3) or the extent to which a license to use any such rights may or may not be available on reasonable and nondiscriminatory terms, or on any terms at all.

© 2013 Network Test Automation Forum

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction other than the following, (1) the above copyright notice and this paragraph must be included on all such copies and derivative works, and (2) this document itself may not be modified in any way, such as by removing the copyright notice or references to NTAF.

By downloading, copying, or using this document in any manner, the user acknowledges that it has read, and hereby consents to, all of the terms and conditions of this notice. Unless the terms and conditions of this notice are breached by the user, the limited permissions granted above are perpetual and will not be revoked by NTAF or its successors or assigns.

**THIS SPECIFICATION AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS BASIS, AND NTAF DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF ANY THIRD PARTY, OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE OR FITNESS FOR A PARTICULAR PURPOSE.**

Revision History

Version	Date	By	Changes
01	2013-08-21	Technical Committee	First release

## Table of Contents

<b>Table of Contents</b> .....	<b>4</b>
<b>Standard Harnesses</b> .....	<b>4</b>
<b>Examples</b> .....	<b>4</b>
<b>Figures</b> .....	<b>4</b>
<b>Works Cited</b> .....	<b>5</b>
<b>1. Introduction</b> .....	<b>5</b>
Motivation.....	5
Inventory .....	5
Activities Using Resources .....	5
Design Goals.....	6
Resource Tracking.....	7
Support Several Test Activities .....	7
Concept .....	7
Silent Resources Supported.....	7
Object Orientation by Unique ID .....	7
<b>2. Discovering Resource Support</b> .....	<b>8</b>
<b>3. Resource Owner Harness</b> .....	<b>8</b>
Resource List.....	8
<b>4. Use Cases</b> .....	<b>9</b>
Find All Resources in a Lab (represented by an NTAF entity) .....	9
Step 1: Find all NTAF entities .....	9
Step 2: Discover entities which support this resource specification .....	9
Step 3: Find resources listed by NTAF entities.....	9

## Standard Harnesses

Harness 1. Resource Owner Harness.....	8
----------------------------------------	---

## Examples

Example 1. Finding all resources .....	9
----------------------------------------	---

## Figures

Figure 1. Network Testing Function Areas.....	6
-----------------------------------------------	---

## Works Cited

- IEEE. (n.d.). *IEEE Standards Association*. Retrieved from <http://standards.ieee.org/develop/regauth/oui/public.html>
- International Organization for Standardization. (n.d.). *ISO/IEC 9834-8:2005*. Retrieved from [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=36775](http://www.iso.org/iso/catalogue_detail.htm?csnumber=36775)
- International Telecommunication Union. (n.d.). *OSI networking and system aspects – Naming, Addressing and Registration*. Retrieved from <http://www.itu.int/ITU-T/studygroups/com17/oid/X.667-E.pdf>
- Network Test Automation Forum. (2010). *NTAF*. Retrieved from Ntaforum: <http://ntaforum.org>
- Network Test Automation Forum. (2011, June). *TS-001: Tool Registration and Discovery*. Retrieved from Ntaforum: <http://ntaforum.org/resources/request.html>
- Network Test Automation Forum. (2011, June). *TS-002: Tool Automation Harness*. Retrieved from Ntaforum: <http://ntaforum.org/resources/request.html>
- Network Working Group. (n.d.). *A Universally Unique Identifier (UUID) URN Namespace*. Retrieved from <http://tools.ietf.org/html/rfc4122>
- Wikipedia. (n.d.). *Organizationally Unique Identifiers*. Retrieved from [http://en.wikipedia.org/wiki/Organizationally\\_unique\\_identifier](http://en.wikipedia.org/wiki/Organizationally_unique_identifier)

## 1. Introduction

This document describes an NTAF (Network Test Automation Forum, 2010) extension that allows tools to communicate resource data. It is based on TS-001 (Network Test Automation Forum, 2011) and TS-002 (Network Test Automation Forum, 2011).

### Motivation

#### Inventory

TS-001 (Network Test Automation Forum, 2011) and TS-002 (Network Test Automation Forum, 2011) together define a basic framework for tools to communicate with one another. Now that such a framework exists it is natural to begin specifying some of the data that will be passed between tools. This document is primarily motivated by the need for labs to manage their inventory. Inventory in this context can be defined as the ability to discover resources and to find relevant resources for a given purpose. To support lab inventory needs this standard seeks to provide an NTAF-based definition of what constitutes a resource and how data about resources is communicated.

#### Activities Using Resources

While inventory is the prime motivation of this standard it is recognized that a good resource standard will also be useful for many other lab activities. For reference, Figure 1 shows the eight functional areas NTAF has defined as covering the field of network testing.

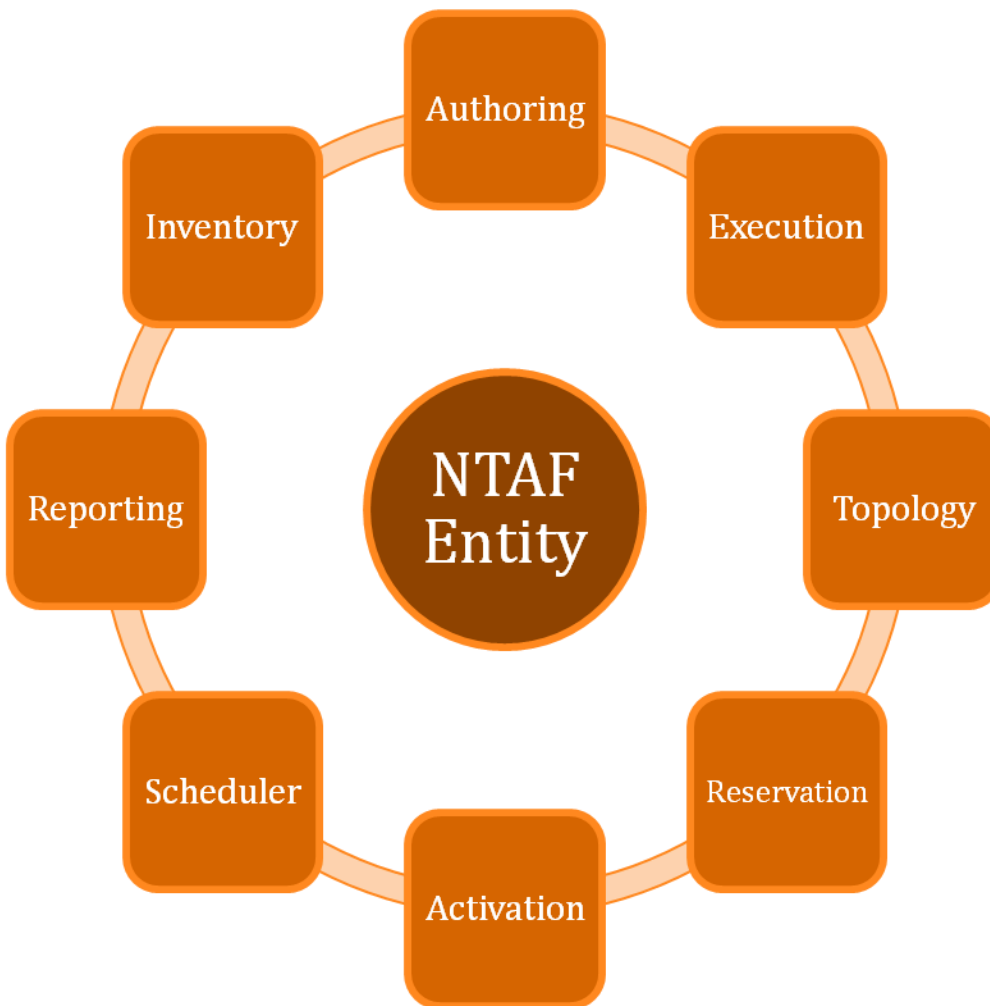
The following listed activities listed are expected to interact with this resource standard.

1. Inventory
2. Reservation

3. Topology
4. Authoring

Inventory has already been mentioned as the primary motivation for this standard. Reservation needs to understand resources to determine when they are in use, mark them as being used or unused, and understand which resources fit the needs of a given resource request. Topology is basically a network of connected or related resources. The test cases produced by authoring are based on resources. Activation has already been defined in TS-001. Scheduling, Execution, and Reporting were left off the list. Scheduling and Execution obviously must use resources, but probably rely on a reservation system for their resource needs rather than referring to them directly. Reporting will typically refer to resources which it has received without directly using the resource standard.

Figure 1. Network Testing Function Areas



## Design Goals

To support inventory and other network test activities this resource standard seeks to meet several design goals

1. Provide a system for communicating information about resources.
2. Provide a method for tracking resources including those which are not NTAF entities but are known by NTAF entities.
3. Provide a resource standard that is useful not only for inventory, but also for reservation, topology, authoring, and activation.

## Resource Tracking

Tools which follow this standard will obviously be NTAF aware entities. This standard seeks to provide a way for such entities to identify themselves as resources. However, many NTAF entities know about resources outside of themselves. For example, a server may know about storage devices or communications channels. Traffic generation software running on one client may be aware of different chassis and the cards they contain. This standard will provide a way for NTAF entities to describe themselves and any other resources of which they are aware. If more than one NTAF entity represents an actual resource it should not introduce an alias for that resource ID. This is to ensure that an inventory system would not count a single resource multiple times.

## Support Several Test Activities

As stated previously, there are many test activities which use resources. This standard seeks to provide resource definitions which are useful to those other activities. It seeks to be flexible not only to provide for future expansion of inventory support, but also to support reservation, topology, authoring, and activation. In particular it is expected that inventory support will come from developing specific interfaces, actions and parameters that can be applied to resources. An inventory system will look for resources that support inventory standards and use the inventory standard to glean information from said resources.

## Concept

### Silent Resources Supported

This resource standard recognizes that NTAF entities often know about resources that are not NTAF enabled. These **silent resources** are of interest in many situations, not the least of which is inventory data gathering. This standard provides a mechanism for communicating about such silent resources. An NTAF entity which supports this standard must be able to list the resources it controls. The list may include itself in addition to any silent resources. The NTAF entity must be able to respond to harness based queries about those resources.

It is possible that multiple NTAF entities list the same resource. For example, traffic generator software may list all the chassis it can connect to. Another instance of the same traffic generator software may list some or all of the same chassis. To avoid confusion, all resources will be identified by a globally unique identifier, as described in the next section.

### Object Orientation by Unique ID

This resource standard will take an object oriented approach toward resource communication. Text-based systems like the XML upon which TS-001 and TS-002 are based do not directly support objects. However, object support is easily simulated by using handles or identifiers to refer to objects. This standard will use globally unique identifiers to refer to resources. These identifiers can be considered as handles to resource objects. The resource list returned by an NTAF entity will be a list of unique identifiers.

Every resource will have an identifier assigned by the NTAF entity that reports it. The identifier must be global – that is it must not change no matter which NTAF entity is reporting it. As such, it must somehow either be innate to the resource or derived from the resource in a consistent manner.

This document specifically declines to state a methodology for generating unique identifiers for resources. Most vendors already maintain uniqueness of identifying information for their products. It does not seem a major change to require this uniqueness for all resources and to assume that vendors can determine a methodology for uniquely identifying their equipment. For example, IEEE has established an Organizationally Unique Identifier (OUI) which can be purchased (IEEE). The OUI serves as a basis for providing an Extended Unique Identifier (EUI) for any product. Both are described in the Wikipedia article on OUIs (Wikipedia). Other examples include RFC 4122 (Network Working Group) (and the matching ITU-T Rec. X.667 (International Telecommunication Union) and ISO/IEC 9834-8 (International Organization for Standardization)) which provide ways of generating unique identifiers on-the-fly which should cover uniquely identifying virtual resources. Again, these are only examples and it is left to each NTAF provider to determine how to uniquely identify its resources.

Resource unique identifiers should conform to the following rules.

- Resource IDs must begin with a letter

- Resource IDs can only contain letters, numbers and the underscore ("\_") character
- Resource IDs are case sensitive. NTAF reserves the right to use characters besides letters, numbers and underscores for special use in future standards. That is, if a resource ID contains such a character, it will be by special provision in a future standard that defines its use.

Unique identifiers are easier to understand for hardware than for software. Nonetheless, unique identifiers still are sensible for software instances. First there is software, or firmware, so closely associated with a piece of hardware that there is no need to identify it apart from the hardware it runs on. The firmware on a router is an example of such software. A second software situation is licensed software. A license tracks its software instances and it may be reasonably inferred that licensed software instances can be uniquely identified by their license. Finally, there is open, unlicensed software. Already, however, TS-001 specified that every NTAF entity must supply a unique workspace ID for its host. It does not seem too great a burden for open software to derive instance uniqueness based on the host workspace ID. This is not a required methodology, only a suggestion. Software may uniquely identify itself any way it chooses.

## 2. Discovering Resource Support

To determine if a given NTAF entity supports resources, one uses the tool registration mechanism described in TS-001. If the entity supports resources, it will support the `http://ntaforum.org/2013/harness/resource/owner` harness described in the following section.

## 3. Resource Owner Harness

### Resource List

Harness 1 shows the Resource Owner Harness that must be supported by any NTAF entity which controls resources. The entity must be able to provide a list of resources it owns through the “getResources” action. The list is a list of identifiers. Each identifier is a string that uniquely identifies its associated resource. This identifier must be consistent even if a resource moves. For example, if a card is unplugged from one chassis and moved to another, it must be reported by that second chassis with the same identifier as that used by the first chassis.

It is assumed this harness will be used for discovery and inventory. To this end it is normally expected that entities will only report resources which they control. If at times there is overlap, it can be detected because resource identifiers are unique. An inventory system can be expected to compare identifiers and notice when there is overlap.

### Harness 1. Resource Owner Harness

```
<query-harness xmlns='http://ntaforum.org/2011/harness'
  harness='http://ntaforum.org/2013/harness/resource/owner'
  xml:lang='en'>
  <label>Resource Owner</label>
  <tooltip>Resource owner interface</tooltip>
  <description>Basic interface for NTAF entities which own resources.</description>
  <author>NTAF WT-003</author>
  <actionDecl name='getResources'>
    <label>Get Resource List</label>
    <tooltip>Fetch the resources owned by this entity</tooltip>
    <description>Fetch the resources owned by this entity. The action must return the entity
  itself. In addition it may return identifiers for additional resources it controls or has
  created. Resources returned are first level children unless the special value "ALL" is sent as
  input.</description>
    <parameter name='resource'>
      <label>Resource</label>
      <tooltip>Identifier for a resource</tooltip>
      <description>An identifier for a resource. The identifier is unique and must be one of
  the identifiers returned by getResources. The special value "SELF" returns resources owned by the
```



```

NTAF entity. The special value "ALL" returns children of the resource to all sub-levels (i.e.
children, grandchildren, etc.).</description>
  <mandatory>false</mandatory>
  <datatype>string</datatype>
  <default>SELF</default>
</parameter>
<responseDecl>
  <item name='resource'>
    <label>Resource ID</label>
    <tooltip>Resource identifier</tooltip>
    <description>Each item is an identifier for a resource. Together the items form a list of
resource identifiers. Resource identifiers must be unique.</description>
    <datatype>string</datatype>
    <allowedCount>
      <min>1</min>
    </allowedCount>
  </item>
</responseDecl>
</actionDecl>
</query-harness>

```

## 4. Use Cases

The notions presented in the previous sections are very generic. It is helpful to examine them and flesh them out in the context of real world use cases.

### Find All Resources in a Lab (represented by an NTAF entity)

Consider the problem of an inventory system finding all the resources in a lab. Using TS-001, TS-002 and this standard the following procedure could be followed.

- Step 1: Find all NTAF entities.
- Step 2: Discover entities which support this resource specification.
- Step 3: Find resources listed by NTAF entities

Now let's examine each step in further detail.

#### Step 1: Find all NTAF entities

TS-001 describes the NTAF registry and how all NTAF entities can be found. Furthermore, using standard XMPP protocol, it is straight forward to determine which of these entities are currently present and able to respond.

#### Step 2: Discover entities which support this resource specification

Among the NTAF entities found, those that support the resource standard will list the `http://ntaforum.org/2013/harness/resource/owner` harness as a supported harness. All entities which support the `http://ntaforum.org/2013/harness/resource/owner` harness can be queried about their resources.

#### Step 3: Find resources listed by NTAF entities

For every NTAF entity that supports `http://ntaforum.org/2013/harness/resource/owner`, the inventory system uses the `getResources` action to find all resources. Using these steps the inventory system has compiled a list of all resources. Example 1 shows some sample transactions.

##### Example 1. Finding all resources

```

<!-- A query to a router which reports only itself as a resource -->
<iq type='get' to='router@domain/1' from='inventorySystem@domain/1' id='find1'>
  <request xmlns='http://ntaforum.org/2011/harness' session='router1'>
    <action harness='http://ntaforum.org/2013/harness/resource/owner'>getResources</action>
  </request>
</iq>

<iq type='result' from='router@domain/1' to='inventorySystem@domain/1' id='find1'>

```

```
<response xmlns='http://ntaforum.org/2011/harness' session='router1'>
  <result>pass</result>
  <item name='resource'>router1</item>
</response>
</iq>

<!-- A query to a modern traffic generator system which controls many resources -->
<iq type='get' to='trafficGenerator@domain/1' from='inventorySystem@domain/1' id='find2'>
  <request xmlns='http://ntaforum.org/2011/harness' session='tg1'>
    <action harness='http://ntaforum.org/2013/harness/resource/owner'>getResources</action>
  </request>
</iq>

<iq type='result' from='trafficGenerator@domain/1' to='inventorySystem@domain/1' id='find2'>
  <response xmlns='http://ntaforum.org/2011/harness' session='tg1'>
    <result>pass</result>
    <item name='resource'>tg1</item>
    <item name='resource'>chassis1</item>
    <item name='resource'>chassis2</item>
    <item name='resource'>router1</item>
    <item name='resource'>router2</item>
  </response>
</iq>
```