

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

**DNS Protocol Modules for TTCN-3 Toolset with TITAN,
User Guide**

Contents

1	Introduction.....	2
1.1	Revision history	2
1.2	About this Document	2
1.2.1	How to Read this Document.....	2
1.2.2	Presumed Knowledge	2
1.2.3	References	2
1.2.4	Abbreviations.....	3
1.2.5	Terminology.....	3
1.3	System Requirements	3
2	Protocol Modules.....	3
2.1	Overview	3
2.2	Installation	4
2.3	Configuration	4
3	Error messages	5
4	Warning messages.....	5
5	Examples.....	6

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

1 Introduction

1.1 Revision history

1.2

Date	Rev	Characteristics	Prepared
2005-04-14	PA1	First draft version	EGBOTAT
2005-04-28	A	Final version after review	EGBOTAT
2005-05-31	B	New RRs added (SRV, NAPTR RR)	EJMTCSO
2006-12-20	PC1	Updated for TITAN R7	ETHBAAT
2007-08-07	PD1	Extended with DNS Update	QATTFLO
2009-03-19	PE1	New RR added (AAAA)	ETHEKR

About this Document

1.2.1 How to Read this Document

This is the User Guide for the set of DNS protocol modules. DNS protocol modules are developed for the TTCN-3 Toolset with TITAN. This document should be read together with Product Revision Information [3] and Function Specification [4].

1.2.2 Presumed Knowledge

To use this protocol module the knowledge of the TTCN-3 core-language [1] and the DNS protocol described in RFC 1035 [5] is essential.

1.2.3 References

- [1] ETSI ES 201 873-1 v3.1.1 (2005-06)
The Testing and Test Control Notation version 3; Part 1: Core Language
- [2] 2/198 17-CRL 113 200 Uen
Programmer's Technical Reference for TITAN TTCN-3 Test Executor
- [3] 109 21-CNL 113 429-1 Uen Rev. B
DNS Protocol Modules for TTCN-3 Toolset with TITAN, Product Revision Information
- [4] 155 17-CNL 113 429 Uen Rev. B
DNS Protocol Modules for TTCN-3 Toolset with TITAN, Function Specification
- [5] [RFC 1035](#)
Domain names – Implementation and specification
- [6] [RFC 2782](#)
A DNS RR for specifying the location of services (DNS SRV)
- [7] [RFC 3403](#)

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

Dynamic Delegation Discovery System (DDDS), Part Three: The Domain Name System (DNS) Database

- [8] [RFC 2136](#)
Dynamic Updates in the Domain Name System (DNS UPDATE)
- [9] [RFC 1886](#)
DNS Extensions to support IP version 6

1.2.4 Abbreviations

ASP	Abstract Service Primitive
DDDS	Dynamic Delegation Discovery System
DNS	Domain Name System
DNS RR	Domain Name System Resource Record
PDU	Protocol Data Unit
RFC	Request For Comments
TTCN-3	Testing and Test Control Notation version 3

1.2.5 Terminology

No specific terminology is used.

1.3 System Requirements

The set of DNS protocol modules are a set of source code files that can only be used as part of a TTCN-3 test suite. Hence, protocol modules alone do not put specific requirements on the system used. However in order to compile and execute a TTCN-3 test suite using a set of protocol modules the following system requirements must be satisfied:

- TITAN TTCN-3 Test Executor R7A (1.7.pl0) or higher installed. For installation guide see [2]. Please note: This version of the protocol module is not compatible with TITAN releases earlier than R7A.

2 Protocol Modules

2.1 Overview

Protocol modules implement the message structure of the related protocol in a formalized way, using the standard specification language TTCN-3. This allows definition of test data (templates) in TTCN-3 core-language [1] and correct encoding/decoding of messages during execution of test suites using the TITAN TTCN-3 test environment.

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

The DNS protocol module implements the following functions:

```
dec_PDU_DNS(in octetstring stream) return PDU_DNS;
```

```
enc_PDU_DNS(in PDU_DNS msg, in boolean doCompression, in  
boolean autoLengthCalc) return octetstring;
```

The function `dec_PDU_DNS` decodes an `octetstring` that represents an encoded DNS message. Unsupported resource records (see Function Specification [4] clause 3.2.1) will be decoded into field `unsupported` and erroneous resource records will be decoded into field `undecodable` of `rData`, both fields are of type `octetstring`. These fields can also be used to encode erroneous or unsupported data. Decoded DNS messages are in uncompressed format.

The function `enc_PDU_DNS` encodes a DNS message given as TTCN-3 data into an `octetstring`. The latter function has two additional `boolean` parameters. Parameter `doCompression` specifies whether the encoder should compress domain names or encode them uncompressed. Parameter `autoLengthCalc` specifies whether the encoder should calculate and substitute length fields in resource records automatically. Calculating length fields manually is not recommended unless it is required to encode a faulty DNS message. Fields `qdCount`, `anCount`, `nsCount` and `arCount` of DNS messages are not automatically calculated (note: if resource records are passed to templates as parameters then `sizeof` can be used to calculate these).

2.2 Installation

The Test Port and the protocol module can be used for the development of TTCN-3 test suites. Since the DNS protocol module is used as part of a TTCN-3 test suite, this requires TTCN-3 Test Executor to be installed before the module can be compiled and executed together with other parts of the test suite. For more details on the installation of TTCN-3 Test Executor see the relevant section of [2].

The followings are needed to use the DNS protocol module:

- Copy the source file (`DNS_EncDec.cc`) and the definition module (`DNS_Types.ttcn`) into the directory of the test suite or create symbolic links to them.
- Import module `DNS_Types` to the module(s) where the DNS protocol module is used.).
- Create a `Makefile` or modify the existing one.

2.3 Configuration

None.

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

3 Error messages

The following error messages can occur while using the DNS protocol module. The error messages are self-explanatory:

Error decoding DNS header. Stream doesn't contain enough octets.

Error decoding query resource record: not enough octets.

Error decoding label at octet *#n*: label length is greater than 63, but is not a pointer (*0xXX*).

Label pointer at octet *#n* in domain name refers after end of stream.

Error decoding label: not enough octets. Remaining bytes: *n*, length of label: *m*

Error decoding resource record: not enough octets. Resource record starts at octet *#n*, number of octets in message: *m*.

Error encoding DNS header: field 'z' must have a length of 3.

Domain name should not start with a dot: "...".

Domain name should not end with a dot: "...".

Label in domain name is longer than 63 characters: "...".

4 Warning messages

The following warning messages can occur while using the DNS protocol module. The warning messages are self-explanatory:

While encoding PDU_DNS: 'header.qdCount' (*n*) differs from size of 'queries' (*m*).

While encoding PDU_DNS: 'header.anCount' (*n*) differs from size of 'answers' (*m*).

While encoding PDU_DNS: 'header.nsCount' (*n*) differs from size of 'nameServerRecords' (*m*).

While encoding PDU_DNS: 'header.arCount' (*n*) differs from size of 'additionalRecords' (*m*).

Forward reference in compressed domain name at octet *#n*.

While decoding resource record: not enough octets, decoding to field 'undecodable'. Resource record starts at octet *#n*.

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

Resource record class n is not supported. rData will be decoded to field 'unsupported'.

While decoding address resource record: not enough octets, decoding to field 'undecodable'.

While decoding SOA resource record: not enough octets, decoding to field 'undecodable'.

While decoding WKS resource record: not enough octets, decoding to field 'undecodable'.

While decoding WKS resource record: rdLength is less than 6. Decoding to field 'undecodable'.

While decoding HINFO resource record: not enough octets, decoding to field 'undecodable'.

While decoding MX resource record: not enough octets, decoding to field 'undecodable'.

While decoding AAAA resource record: not enough octets, decoding to field 'undecodable'.

While decoding SRV resource record: not enough octets, decoding to field 'undecodable'.

While decoding NAPTR resource record: not enough octets, decoding to field 'undecodable'.

Resource record type n is not supported. RData will be decoded to field 'unsupported'.

While decoding resource record: rdLength (n) does not equal the length of decoded resource record data (m). Resource record starts at octet $\#k$.

Length of rData (n) in octets differs from rdLength (m) in resource record: ...

While encoding domain name: domain name is empty.

In case of Resource record class n rdLength and ttl fields must be zero. rData will be decoded to field 'unsupported'.

5 Examples

The "demo" directory of the deliverable contains the following examples:

- `DNS_Types.ttcn`: definition module.
- `DNS_EncDec.cc`: source file.
- `DNS_Demo.ttcn`: demo testcases for DNS protocol module.

Prepared (also subject responsible if other) ETH/XZX Endre Kulcsár +36 1 437 7469		No. 198 17-CNL 113 429 Uen		
Approved ETH/XZXC (Tibor Csondes)	Checked	Date 2011-11-29	Rev E	Reference GASK2

- `DNS_Mapping.ttcn`: mapping module.
- `Makefile`
- `UDPasp_PortType.ttcn`: port type definition for UDP test port (CNL 113 346).
- `UDPasp_Types.ttcn`: type definitions for UDP test port.
- `UDPasp_PT.cc`: source file of UDP test port.
- `UDPasp_PT.hh`: header file of UDP test port.
- `config.cfg`: example runtime configuration file.
- `readme.txt`: brief description of how to use the demo.