Registry System Testing

Whois Test Area Specification

Version ~~C~~D

File name: Test Area Whois.docx
Last saved: 2021-06-09

Copyright (c) 2017-2018 Internet Corporation for Assigned Names and Numbers. All rights reserved.

Document control

Document information and security

Made by	Responsible for fact	Responsible for document
Lennart Bonnevier	Mats Dufberg	Mats Dufberg

Security class	File name
External	Test Area Whois.docx

Revisions

Date	Version	Name	Description
2017-07-27	A	Mats Dufberg	First release version.
2017-07-28	B	Mats Dufberg	Added missing information on SRS GW test nodes and IPv4/IPv6.
2018-06-27	C	Mats Dufberg	Added references to "Temporary Specification for gTLD Registration Data". Updated the format specification in the Appendix to meet the Temporary Specification. Directly or indirectly adjusted Test Cases WhoisCLI01, WhoisWeb03, WhoisSearch02 and SRS+GWWhoisCLI01 to the Temporary Specification. Added full test case definition of WhoisWeb09 which was deleted by mistake in earlier version.
2021-06-10	D	Mauro Lozano	Update references.

LIST OF CONTENTS

1. INTRODUCTION 14
1.1 Scope 14
1.2 References 14
1.2.1 External 14
1.2.2 Standards 14
1.2.3 Repositories 15
1.2.4 Other references 15
1.2.5 Internal 15
1.2.6 Document Hierarchy 16
1.3 Earlier documents 16
1.4 Level in the overall sequence 16
1.5 Test classes and overall test conditions 16
1.6 Special test conditions for SRS Gateway Testing 16
2. TEST REQUIREMENTS 18
2.1 Test items and their identifiers 18
2.1.1 Statement of Work 18
2.1.2 Applicant Guidebook 18
2.1.3 Specification 4 19
2.1.4 RDDS Advisory, ROID Advisory, AWIP, RDDS Consistent Labeling Policy, and Temp Spec for gTLD data 19
2.1.5 EPP 20
2.1.6 RFC 21
2.1.7 SRS Gateway testing 21
2.2 Features to be tested 21
2.3 Features not to be tested 21
2.4 Approach 21
2.5 Item pass/fail criteria 22
2.6 Suspension criteria and resumption requirements 22
2.7 Test deliverables 22
3. TEST TRACEABILITY MATRIX 23
4. TEST MANAGEMENT 25
5. TEST CASE WHOISCLI01: QUERY FOR DOMAIN NAME 26
5.1 Test case identifier 26
5.2 Objective 26
5.3 Inputs 26
5.4 Outcome(s) 26
5.5 Environmental needs 26
5.6 Special procedural requirements 26
5.7 Intercase dependencies 26
5.8 Ordered description of steps to be taken to execute the test case 26
6. TEST CASE WHOISCLI02: QUERY FOR REGISTRAR 28
6.1 Test case identifier 28
6.2 Objective 28
6.3 Inputs 28
6.4 Outcome(s) 28
6.5 Environmental needs 28
6.6 Special procedural requirements 28
6.7 Intercase dependencies 28
6.8 Ordered description of steps to be taken to execute the test case 28
7. TEST CASE WHOISCLI03: QUERY FOR NAME SERVER 30

7.1 Test case identifier 30
7.2 Objective 30
7.3 Inputs 30
7.4 Outcome(s) 30
7.5 Environmental needs 30
7.6 Special procedural requirements 30
7.7 Intercase dependencies 30
7.8 Ordered description of steps to be taken to execute the test case 31
8. TEST CASE WHOISWEB01: VERIFY IPV4 CONNECTIVITY 33
8.1 Test case identifier 33
8.2 Objective 33
8.3 Inputs 33
8.4 Outcome(s) 33
8.5 Environmental needs 33
8.6 Special procedural requirements 33
8.7 Intercase dependencies 33
8.8 Ordered description of steps to be taken to execute the test case 33
9. TEST CASE WHOISWEB02: VERIFY IPV6 CONNECTIVITY 35
9.1 Test case identifier 35
9.2 Objective 35
9.3 Inputs 35
9.4 Outcome(s) 35
9.5 Environmental needs 35
9.6 Special procedural requirements 35
9.7 Intercase dependencies 35
9.8 Ordered description of steps to be taken to execute the test case 35
10. TEST CASE WHOISWEB03: MANUAL QUERY FOR DOMAIN NAME 37
10.1 Test case identifier 37
10.2 Objective 37
10.3 Inputs 37
10.4 Outcome(s) 37
10.5 Environmental needs 37
10.6 Special procedural requirements 37
10.7 Intercase dependencies 37
10.8 Ordered description of steps to be taken to execute the test case 37
11. TEST CASE WHOISWEB04: MANUAL QUERY FOR REGISTRAR 39
11.1 Test case identifier 39
11.2 Objective 39
11.3 Inputs 39
11.4 Outcome(s) 39
11.5 Environmental needs 39
11.6 Special procedural requirements 39
11.7 Intercase dependencies 39
11.8 Ordered description of steps to be taken to execute the test case 39
12. TEST CASE WHOISWEB05: MANUAL QUERY FOR NAME SERVER 41
12.1 Test case identifier 41
12.2 Objective 41
12.3 Inputs 41
12.4 Outcome(s) 41
12.5 Environmental needs 41
12.6 Special procedural requirements 41
12.7 Intercase dependencies 41
12.8 Ordered description of steps to be taken to execute the test case 41

13. WHOIS WEB 09: MANUAL QUERY OVER IPV6 43
13.1 Test case identifier 43
13.2 Objective 43
13.3 Inputs 43
13.4 Outcome(s) 43
13.5 Environmental needs 43
13.6 Special procedural requirements 43
13.7 Intercase dependencies 43
13.8 Ordered description of steps to be taken to execute the test case 43
14. TEST CASE WHOISSEARCH00: VERIFY IF WHOIS SEARCH IS SUPPORTED 45
14.1 Test case identifier 45
14.2 Objective 45
14.3 Inputs 45
14.4 Outcome(s) 45
14.5 Environmental needs 45
14.6 Special procedural requirements 45
14.7 Ordered description of steps to be taken to execute the test case 45
15. TEST CASE WHOISSEARCH01: VERIFY ABUSE PROTECTION 47
15.1 Test case identifier 47
15.2 Objective 47
15.3 Inputs 47
15.4 Outcome(s) 47
15.5 Environmental needs 47
15.6 Special procedural requirements 47
15.7 Intercase dependencies 47
15.8 Ordered description of steps to be taken to execute the test case 47
16. TEST CASE WHOISSEARCH02: PARTIAL MATCH CAPABILITIES 49
16.1 Test case identifier 49
16.2 Objective 49
16.3 Inputs 49
16.4 Outcome(s) 49
16.5 Environmental needs 49
16.6 Special procedural requirements 49
16.7 Intercase dependencies 49
16.8 Ordered description of steps to be taken to execute the test case 49
17. TEST CASE WHOISSEARCH03: EXACT MATCH SEARCH 51
17.1 Test case identifier 51
17.2 Objective 51
17.3 Inputs 51
17.4 Outcome(s) 51
17.5 Environmental needs 51
17.6 Special procedural requirements 51
17.7 Intercase dependencies 51
17.8 Ordered description of steps to be taken to execute the test case 51
18. TEST CASE WHOISSEARCH04: BOOLEAN SEARCH CAPABILITIES 53
18.1 Test case identifier 53
18.2 Objective 53
18.3 Inputs 53
18.4 Outcome(s) 53
18.5 Environmental needs 53
18.6 Special procedural requirements 53
18.7 Intercase dependencies 53
18.8 Ordered description of steps to be taken to execute the test case 53

19. TEST CASE WHOISSEARCH09: SEARCH OVER IPV6 55
19.1 Test Case Identifier 55
19.2 Objective 55
19.3 Inputs 55
19.4 Outcome(s) 55
19.5 Environmental needs 55
19.6 Special procedure requirements 55
19.7 Intercase dependencies 55
19.8 Ordered description of steps to be taken to execute the test case 55
20. TEST CASE SRS+GW WHOIS CLI 01: VERIFY CONSISTENCY FOR DOMAIN NAME OBJECTS 57
20.1 Test case identifier 57
20.2 Objective 57
20.3 Inputs 57
20.4 Outcome(s) 57
20.5 Environmental needs 57
20.6 Special procedural requirements 58
20.7 Intercase dependencies 58
20.8 Ordered description of steps to be taken to execute the test case 58
21. TEST CASE SRS+GW WHOIS CLI 02: VERIFY CONSISTENCY FOR REGISTRAR OBJECTS 59
21.1 Test case identifier 59
21.2 Objective 59
21.3 Inputs 59
21.4 Outcome(s) 59
21.5 Environmental needs 59
21.6 Special procedural requirements 60
21.7 Intercase dependencies 60
21.8 Ordered description of steps to be taken to execute the test case 60
22. TEST CASE SRS+GW WHOIS CLI 03: VERIFY CONSISTENCY FOR NAME SERVER OBJECTS 61
22.1 Test case identifier 61
22.2 Objective 61
22.3 Inputs 61
22.4 Outcome(s) 61
22.5 Environmental needs 61
22.6 Special procedural requirements 62
22.7 Intercase dependencies 62
22.8 Ordered description of steps to be taken to execute the test case 62
23. GENERAL 64
23.1 Glossary 64
23.2 Document change procedures 64
24. APPENDIX: FORMAT SPECIFICATION 65
24.1 Terminology 65
24.2 Limitation 65
24.3 Overview 65
24.4 Characters and line structure 65
24.5 Section types 66
24.5.1 Whois output 66
24.5.2 Registrar reply 67
24.5.3 Domain name reply 67
24.5.4 Name server reply type 1 67
24.5.5 Name server reply type 2 67

24.5.6 Subsequent registrar details section 67
24.5.7 Registrar details section 67
24.5.8 Admin contact section 68
24.5.9 Technical contact section 68
24.5.10 Phone number section 68
24.5.11 Fax number section 68
24.5.12 Fax number section type A 68
24.5.13 Fax number section type B 68
24.5.14 Fax number section type C 68
24.5.15 Subsequent domain name details section 69
24.5.16 Domain name details section 69
24.5.17 Domain name details subsection 1 69
24.5.18 Domain name details subsection 2 69
24.5.19 Domain name details subsection 3 69
24.5.20 Domain name details subsection 4 69
24.5.21 Domain abuse contact section 70
24.5.22 Billing contact section 70
24.5.23 Name server section 71
24.5.24 Name server section type A 71
24.5.25 Name server section type B 71
24.5.26 Name server section type C 71
24.5.27 IP Address section 71
24.5.28 Multiple name servers section 71
24.5.29 Subsequent name server details section 71
24.5.30 Name server details section 71
24.5.31 Registrar object additional fields section 71
24.5.32 Registrar object additional field 71
24.5.33 Domain name object additional fields section 72
24.5.34 Domain name object additional field 72
24.5.35 Name server object additional fields section 72
24.5.36 Name server object additional field 72
24.5.37 Last update footer 72
24.5.38 AWIP footer 72
24.5.39 Legal disclaimer 72
24.6 LINE TYPES 72
24.6.1 Empty line 72
24.6.2 Non-empty line 73
24.6.3 Non-empty field 73
24.6.4 Empty field 73
24.6.5 Multiple name servers line 73
24.6.6 ROID line 73
24.6.7 Last update line 73
24.6.8 AWIP line 74
24.7 QUANTIFIERS 74
24.7.1 Required 74
24.7.2 Optional-constrained 74
24.7.3 Empty-constrained 74
24.7.4 Omitted-constrained 74
24.7.5 Special restrictions on "constrained" 75
24.7.6 Optional-free 75
24.7.7 Optional-not-empty 75
24.7.8 Repeatable 75
24.7.9 Optional-repeatable 75
24.8 DATA TYPES 76
24.8.1 Translation clause 76
24.8.2 Key translation 76
24.8.3 Field key 76
24.8.4 Redact String 76
24.8.5 ROID 76

~~24.8.6 ROID Suffix~~ 76
~~24.8.7 ROID or redacted~~ 76
~~24.8.8 Hostname~~ 76
~~24.8.9 Time stamp~~ 77
~~24.8.10 U-label~~ 77
~~24.8.11 HTTP URL~~ 77
~~24.8.12 Token~~ 77
~~24.8.13 Token or redacted~~ 77
~~24.8.14 Positive integer~~ 77
~~24.8.15 Domain status~~ 77
~~24.8.16 Domain status code~~ 78
~~24.8.17 Postal line~~ 78
~~24.8.18 Postal code~~ 78
~~24.8.19 Postal code or redacted~~ 78
~~24.8.20 Country code~~ 78
~~24.8.21 Country code or redacted~~ 79
~~24.8.22 Phone number~~ 79
~~24.8.23 Phone number or redacted~~ 79
~~24.8.24 Email address~~ 79
~~24.8.25 Email redact string~~ 79
~~24.8.26 Email web or redacted~~ 79
~~24.8.27 DNSSEC~~ 79
~~24.8.28 IP address~~ 79
~~24.8.29 IPv4 address~~ 80
~~24.8.30 IPv6 address~~ 80
~~24.8.31 Void~~ 80
~~24.8.32 Forbidden Registrar object additional field names~~ 80
~~24.8.33 Forbidden Domain name object additional field names~~ 81
~~24.8.34 Forbidden Name server object additional field names~~ 82
~~24.8.35 Inaccuracy complaint URL~~ 82

1. INTRODUCTION 14

1.1 SCOPE 14
1.2 REFERENCES 14
1.2.1 External 14
1.2.2 Standards 14
1.2.3 Repositories 15
1.2.4 Other references 15
1.2.5 Internal 15
1.2.6 Document Hierarchy 16
1.3 EARLIER DOCUMENTS 16
1.4 LEVEL IN THE OVERALL SEQUENCE 16
1.5 TEST CLASSES AND OVERALL TEST CONDITIONS 16
1.6 SPECIAL TEST CONDITIONS FOR SRS GATEWAY TESTING 16

2. TEST REQUIREMENTS 18

2.1 TEST ITEMS AND THEIR IDENTIFIERS 18
2.1.1 Applicant Guidebook 18
2.1.2 Specification 4 19
2.1.3 RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy, and Temp-Spec-for-gTLD-data 19
2.1.4 EPP 20
2.1.5 RFC 21
2.1.6 SRS Gateway testing 21
2.2 FEATURES TO BE TESTED 21
2.3 FEATURES NOT TO BE TESTED 21
2.4 APPROACH 21
2.5 ITEM PASS/FAIL CRITERIA 22
2.6 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS 22

2.7 TEST DELIVERABLES 22
3. TEST TRACEABILITY MATRIX 23
4. TEST MANAGEMENT 25
5. TEST CASE WHOISCLI01: QUERY FOR DOMAIN NAME 26
5.1 TEST CASE IDENTIFIER 26
5.2 OBJECTIVE 26
5.3 INPUTS 26
5.4 OUTCOME(S) 26
5.5 ENVIRONMENTAL NEEDS 26
5.6 SPECIAL PROCEDURAL REQUIREMENTS 26
5.7 INTERCASE DEPENDENCIES 26
5.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 26
6. TEST CASE WHOISCLI02: QUERY FOR REGISTRAR 28
6.1 TEST CASE IDENTIFIER 28
6.2 OBJECTIVE 28
6.3 INPUTS 28
6.4 OUTCOME(S) 28
6.5 ENVIRONMENTAL NEEDS 28
6.6 SPECIAL PROCEDURAL REQUIREMENTS 28
6.7 INTERCASE DEPENDENCIES 28
6.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 28
7. TEST CASE WHOISCLI03: QUERY FOR NAME SERVER 30
7.1 TEST CASE IDENTIFIER 30
7.2 OBJECTIVE 30
7.3 INPUTS 30
7.4 OUTCOME(S) 30
7.5 ENVIRONMENTAL NEEDS 30
7.6 SPECIAL PROCEDURAL REQUIREMENTS 30
7.7 INTERCASE DEPENDENCIES 30
7.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 31
8. TEST CASE WHOISWEB01: VERIFY IPV4 CONNECTIVITY 33
8.1 TEST CASE IDENTIFIER 33
8.2 OBJECTIVE 33
8.3 INPUTS 33
8.4 OUTCOME(S) 33
8.5 ENVIRONMENTAL NEEDS 33
8.6 SPECIAL PROCEDURAL REQUIREMENTS 33
8.7 INTERCASE DEPENDENCIES 33
8.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 33
9. TEST CASE WHOISWEB02: VERIFY IPV6 CONNECTIVITY 35
9.1 TEST CASE IDENTIFIER 35
9.2 OBJECTIVE 35
9.3 INPUTS 35
9.4 OUTCOME(S) 35
9.5 ENVIRONMENTAL NEEDS 35
9.6 SPECIAL PROCEDURAL REQUIREMENTS 35
9.7 INTERCASE DEPENDENCIES 35
9.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 35
10. TEST CASE WHOISWEB03: MANUAL QUERY FOR DOMAIN NAME 37
10.1 TEST CASE IDENTIFIER 37
10.2 OBJECTIVE 37
10.3 INPUTS 37

10.4 OUTCOME(S) 37
10.5 ENVIRONMENTAL NEEDS 37
10.6 SPECIAL PROCEDURAL REQUIREMENTS 37
10.7 INTERCASE DEPENDENCIES 37
10.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 37
11. TEST CASE WHOISWEB04: MANUAL QUERY FOR REGISTRAR 39
11.1 TEST CASE IDENTIFIER 39
11.2 OBJECTIVE 39
11.3 INPUTS 39
11.4 OUTCOME(S) 39
11.5 ENVIRONMENTAL NEEDS 39
11.6 SPECIAL PROCEDURAL REQUIREMENTS 39
11.7 INTERCASE DEPENDENCIES 39
11.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 39
12. TEST CASE WHOISWEB05: MANUAL QUERY FOR NAME SERVER 41
12.1 TEST CASE IDENTIFIER 41
12.2 OBJECTIVE 41
12.3 INPUTS 41
12.4 OUTCOME(S) 41
12.5 ENVIRONMENTAL NEEDS 41
12.6 SPECIAL PROCEDURAL REQUIREMENTS 41
12.7 INTERCASE DEPENDENCIES 41
12.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 41
13. WHOIS WEB 09: MANUAL QUERY OVER IPV6 43
13.1 TEST CASE IDENTIFIER 43
13.2 OBJECTIVE 43
13.3 INPUTS 43
13.4 OUTCOME(S) 43
13.5 ENVIRONMENTAL NEEDS 43
13.6 SPECIAL PROCEDURAL REQUIREMENTS 43
13.7 INTERCASE DEPENDENCIES 43
13.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 43
14. TEST CASE WHOISSEARCH00: VERIFY IF WHOIS SEARCH IS SUPPORTED 45
14.1 TEST CASE IDENTIFIER 45
14.2 OBJECTIVE 45
14.3 INPUTS 45
14.4 OUTCOME(S) 45
14.5 ENVIRONMENTAL NEEDS 45
14.6 SPECIAL PROCEDURAL REQUIREMENTS 45
14.7 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 45
15. TEST CASE WHOISSEARCH01: VERIFY ABUSE PROTECTION 47
15.1 TEST CASE IDENTIFIER 47
15.2 OBJECTIVE 47
15.3 INPUTS 47
15.4 OUTCOME(S) 47
15.5 ENVIRONMENTAL NEEDS 47
15.6 SPECIAL PROCEDURAL REQUIREMENTS 47
15.7 INTERCASE DEPENDENCIES 47
15.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 47
16. TEST CASE WHOISSEARCH02: PARTIAL MATCH CAPABILITIES 49
16.1 TEST CASE IDENTIFIER 49
16.2 OBJECTIVE 49
16.3 INPUTS 49

16.4 OUTCOME(S) 49
16.5 ENVIRONMENTAL NEEDS 49
16.6 SPECIAL PROCEDURAL REQUIREMENTS 49
16.7 INTERCASE DEPENDENCIES 49
16.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 49

17. TEST CASE WHOISSEARCH03: EXACT MATCH SEARCH 51

17.1 TEST CASE IDENTIFIER 51
17.2 OBJECTIVE 51
17.3 INPUTS 51
17.4 OUTCOME(S) 51
17.5 ENVIRONMENTAL NEEDS 51
17.6 SPECIAL PROCEDURAL REQUIREMENTS 51
17.7 INTERCASE DEPENDENCIES 51
17.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 51

18. TEST CASE WHOISSEARCH04: BOOLEAN SEARCH CAPABILITIES 53

18.1 TEST CASE IDENTIFIER 53
18.2 OBJECTIVE 53
18.3 INPUTS 53
18.4 OUTCOME(S) 53
18.5 ENVIRONMENTAL NEEDS 53
18.6 SPECIAL PROCEDURAL REQUIREMENTS 53
18.7 INTERCASE DEPENDENCIES 53
18.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 53

19. TEST CASE WHOISSEARCH09: SEARCH OVER IPV6 55

19.1 TEST CASE IDENTIFIER 55
19.2 OBJECTIVE 55
19.3 INPUTS 55
19.4 OUTCOME(S) 55
19.5 ENVIRONMENTAL NEEDS 55
19.6 SPECIAL PROCEDURE REQUIREMENTS 55
19.7 INTERCASE DEPENDENCIES 55
19.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 55

20. TEST CASE SRS+GW WHOIS CLI 01: VERIFY CONSISTENCY FOR DOMAIN NAME OBJECTS 57

20.1 TEST CASE IDENTIFIER 57
20.2 OBJECTIVE 57
20.3 INPUTS 57
20.4 OUTCOME(S) 57
20.5 ENVIRONMENTAL NEEDS 57
20.6 SPECIAL PROCEDURAL REQUIREMENTS 58
20.7 INTERCASE DEPENDENCIES 58
20.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 58

21. TEST CASE SRS+GW WHOIS CLI 02: VERIFY CONSISTENCY FOR REGISTRAR OBJECTS 59

21.1 TEST CASE IDENTIFIER 59
21.2 OBJECTIVE 59
21.3 INPUTS 59
21.4 OUTCOME(S) 59
21.5 ENVIRONMENTAL NEEDS 59
21.6 SPECIAL PROCEDURAL REQUIREMENTS 60
21.7 INTERCASE DEPENDENCIES 60
21.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 60

22. TEST CASE SRS+GW WHOIS CLI 03: VERIFY CONSISTENCY FOR NAME SERVER OBJECTS 61
22.1 TEST CASE IDENTIFIER 61
22.2 OBJECTIVE 61
22.3 INPUTS 61
22.4 OUTCOME(S) 61
22.5 ENVIRONMENTAL NEEDS 61
22.6 SPECIAL PROCEDURAL REQUIREMENTS 62
22.7 INTERCASE DEPENDENCIES 62
22.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 62
23. GENERAL 64
23.1 GLOSSARY 64
23.2 DOCUMENT CHANGE PROCEDURES 64
24. APPENDIX: FORMAT SPECIFICATION 65
24.1 TERMINOLOGY 65
24.2 LIMITATION 65
24.3 OVERVIEW 65
24.4 CHARACTERS AND LINE STRUCTURE 65
24.5 SECTION TYPES 66
24.5.1 Whois output 66
24.5.2 Registrar reply 67
24.5.3 Domain name reply 67
24.5.4 Name server reply type 1 67
24.5.5 Name server reply type 2 67
24.5.6 Subsequent registrar details section 67
24.5.7 Registrar details section 67
24.5.8 Admin contact section 68
24.5.9 Technical contact section 68
24.5.10 Phone number section 68
24.5.11 Fax number section 68
24.5.12 Fax number section type A 68
24.5.13 Fax number section type B 68
24.5.14 Fax number section type C 68
24.5.15 Subsequent domain name details section 69
24.5.16 Domain name details section 69
24.5.17 Domain name details subsection 1 69
24.5.18 Domain name details subsection 2 69
24.5.19 Domain name details subsection 3 69
24.5.20 Domain name details subsection 4 69
24.5.21 Domain abuse contact section 70
24.5.22 Billing contact section 70
24.5.23 Name server section 71
24.5.24 Name server section type A 71
24.5.25 Name server section type B 71
24.5.26 Name server section type C 71
24.5.27 IP Address section 71
24.5.28 Multiple name servers section 71
24.5.29 Subsequent name server details section 71
24.5.30 Name server details section 71
24.5.31 Registrar object additional fields section 71
24.5.32 Registrar object additional field 71
24.5.33 Domain name object additional fields section 72
24.5.34 Domain name object additional field 72
24.5.35 Name server object additional fields section 72
24.5.36 Name server object additional field 72
24.5.37 Last update footer 72

24.5.38 AWIP footer 72
24.5.39 Legal disclaimer 72
24.6 LINE TYPES 72
24.6.1 Empty line 72
24.6.2 Non-empty line 73
24.6.3 Non-empty field 73
24.6.4 Empty field 73
24.6.5 Multiple name servers line 73
24.6.6 ROID line 73
24.6.7 Last update line 73
24.6.8 AWIP line 74
24.7 QUANTIFIERS 74
24.7.1 Required 74
24.7.2 Optional-constrained 74
24.7.3 Empty-constrained 74
24.7.4 Omitted-constrained 74
24.7.5 Special restrictions on "constrained" 75
24.7.6 Optional-free 75
24.7.7 Optional-not-empty 75
24.7.8 Repeatable 75
24.7.9 Optional-repeatable 75
24.8 DATA TYPES 76
24.8.1 Translation clause 76
24.8.2 Key translation 76
24.8.3 Field key 76
24.8.4 Redact String 76
24.8.5 ROID 76
24.8.6 ROID Suffix 76
24.8.7 ROID-or-redacted 76
24.8.8 Hostname 76
24.8.9 Time stamp 77
24.8.10 U-label 77
24.8.11 HTTP URL 77
24.8.12 Token 77
24.8.13 Token-or-redacted 77
24.8.14 Positive integer 77
24.8.15 Domain status 77
24.8.16 Domain status code 78
24.8.17 Postal line 78
24.8.18 Postal code 78
24.8.19 Postal-code-or-redacted 78
24.8.20 Country code 78
24.8.21 Country-code-or-redacted 79
24.8.22 Phone number 79
24.8.23 Phone-number-or-redacted 79
24.8.24 Email address 79
24.8.25 Email redact string 79
24.8.26 Email-web-or-redacted 79
24.8.27 DNSSEC 79
24.8.28 IP address 79
24.8.29 IPv4 address 80
24.8.30 IPv6 address 80
24.8.31 Void 80
24.8.32 Forbidden Registrar object additional field names 80
24.8.33 Forbidden Domain name object additional field names 81
24.8.34 Forbidden Name server object additional field names 82
24.8.35 Inaccuracy complaint URL 82

1. Introduction

This document describes the Whois Level Tests within the Registry System Testing framework.

1.1 Scope

The Registry System Testing (RST) Provider will test the Whois service over port 43 (Whois) and 80/443 (HTTP/HTTPS), and verify the response format. All tests are performed over IPv4 and IPv6 from at least five points on the Internet, unless explicitly stated otherwise. At least one test node is located in every ICANN region.

1.2 References

1.2.1 External

- IEEE 829-2008
- ICANN gTLD Applicant Guidebook, Version 2012-06-04
- RDDS-Advisory, "Advisory: Clarifications to the Registry Agreement, and the 2013 Registrar Accreditation Agreement (RAA) regarding applicable Registration Data Directory Service (Whois) Specifications", 2015-04-27, https://www.icann.org/resources/pages/registry-agreement-raa-rdds-2015-04-27-en
- Registry-Agreement, "Base Registry Agreement", Updated 09 January 2014, http://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-09jan14-en.pdf
- AWIP, "Additional Whois Information Policy", 2014-07-02, https://www.icann.org/resources/pages/policy-awip-2014-07-02-en
- ROID-Advisory, "gTLD Registry Advisory: Correction of non-compliant ROIDs", 2015-08-26, https://www.icann.org/resources/pages/correction-non-compliant-roids-2015-08-26-en
- RDDS-Consistent-Labeling-Policy, "Registry Registration Data Directory Services Consistent Labeling and Display Policy", https://www.icann.org/rdds-labeling-display
- Temp-Spec-for-gTLD-data, "Temporary Specification for gTLD Registration Data", 2018-05-17, https://www.icann.org/resources/pages/gtld-registration-data-specs-en

1.2.2 Standards

- RFC791, "Internet Protocol. Darpa Internet Program Protocol Specification", https://tools.ietf.org/html/rfc791
- RFC952, "DOD Internet Host Table Specification", https://tools.ietf.org/html/rfc952
- RFC1123, "Requirements for Internet Hosts -- Application and Support", https://tools.ietf.org/html/rfc1123
- RFC 2119, "Key words for use in RFCs to Indicate Requirement Levels", https://tools.ietf.org/html/rfc2119
- RFC3339, "Date and Time on the Internet: Timestamps", https://tools.ietf.org/html/rfc3339
- RFC3915, "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP) ", https://tools.ietf.org/html/rfc3915
- RFC4291, "IP Version 6 Addressing Architecture", https://tools.ietf.org/html/rfc4291
- RFC5322, "Internet Message Format", https://tools.ietf.org/html/rfc5322
- RFC5730, "Extensible Provisioning Protocol (EPP)", https://tools.ietf.org/html/rfc5730

- RFC5731, "Extensible Provisioning Protocol (EPP) Domain Name Mapping", https://tools.ietf.org/html/rfc5731
- RFC5732, "Extensible Provisioning Protocol (EPP) Host Mapping", https://tools.ietf.org/html/rfc5732
- RFC5733, "Extensible Provisioning Protocol (EPP) Contact Mapping", https://tools.ietf.org/html/rfc5733
- RFC5734, "Extensible Provisioning Protocol (EPP) Transport over TCP", https://tools.ietf.org/html/rfc5734
- RFC5890, "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", https://tools.ietf.org/html/rfc5890
- RFC7230, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", https://tools.ietf.org/html/rfc7230
- XML-Schema, "XML Schema Part 2: Datatypes Second Edition", http://www.w3.org/TR/xmlschema-2/

1.2.3 Repositories

- IANA-Repository, "Extensible Provisioning Protocol (EPP) Repository Identifiers", https://www.iana.org/assignments/epp-repository-ids/epp-repository-ids.xhtml,

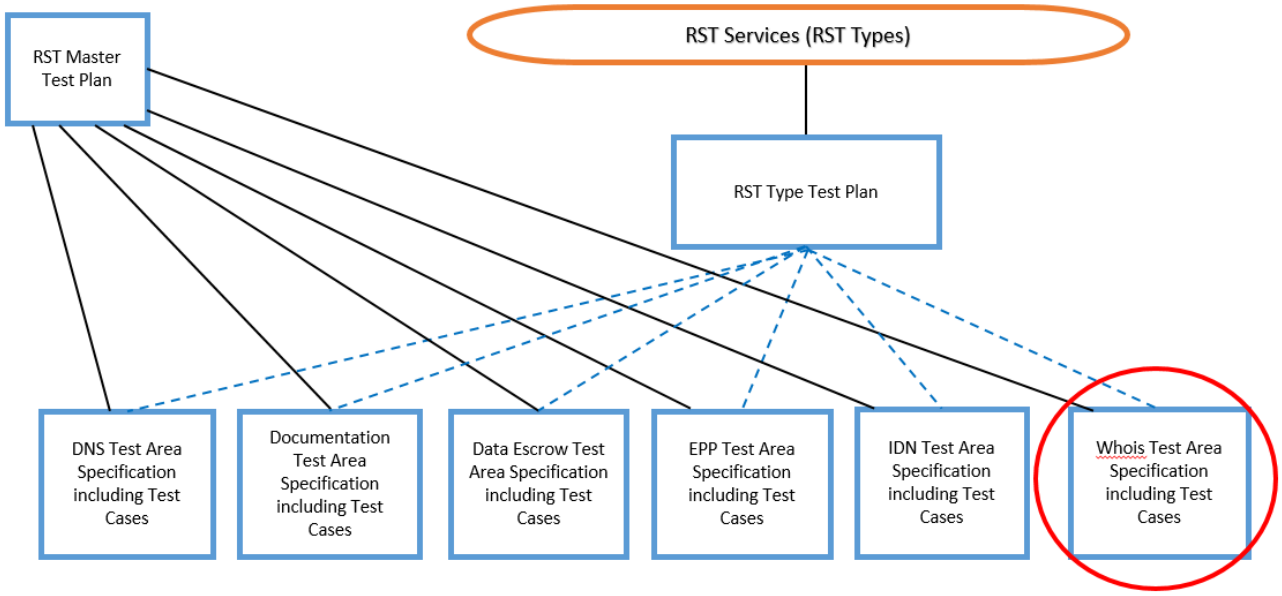
1.2.4 Other references

- Dot-Decimal, "Dot-decimal notation", https://en.wikipedia.org/wiki/Dot-decimal_notation
- ISO3166-1-ISO, "ISO 3166-1 Country Code list" can be searched at https://www.iso.org/
- ISO3166-1-Wikipedia, "ISO 3166-1", https://en.wikipedia.org/wiki/ISO_3166-1

1.2.5 Internal

- ~~Pre-Delegation Testing, Statement of Work~~
- Registry System Testing, Master Test Plan
- Registry System Testing, Documentation Test Area Specification
- ~~SRS Gateway Testing, Statement of Work~~
- Design and principles for SRS Gateway Testing

1.2.6 Document Hierarchy



This document is one of many Test Area Specifications for RST (circled in red in the above graphic). It defines the Test Cases for its Test Area.

1.3 Earlier documents

This document replaces, in contents, the following documents that were part of PDT (Pre-Delegation Testing):

- Pre-Delegation Testing: Whois Test Plan (version K)
- Pre-Delegation Testing: Whois CLI Test Cases (version I)
- Pre-Delegation Testing: Whois Web Test Cases (version H)
- Pre-Delegation Testing: Whois Search Test Cases (version G)
- SRS Gateway Testing: Test Sequence Three Test Cases: validation of consistency in Whois (version A)

1.4 Level in the overall sequence

This Test Area and the associated Test Cases can be run in parallel with the other Test Areas.

1.5 Test classes and overall test conditions

The Whois service of the gTLD is available over IPv4/IPv6 via TCP port 43 and via a web interface. Both reachability and response format will be tested with positive test cases. Valid test data is provided by the Registry Operator. In case of an IDN-TLD, the ASCII-compatible form (A-Label) must be used.

1.6 Special test conditions for SRS Gateway Testing

All SRS Gateway Test Cases (see sections 20, 21 and 22) are performed over IPv4 from test nodes. When possible, tests are also performed over IPv6.

- <TestNodeA>, one test node somewhere in the world, not in the same country as the SRS GW, for the Whois queries to the TLD SRS Whois server. It has always both IPv4 and IPv6 connectivity.
- <TestNodeB>, one test node in the same country as the SRS GW Whois server for the Whois queries to that Whois server. It has always IPv4 connectivity, and it may have IPv6 connectivity.

The IP addresses to the Whois servers are submitted by the Registry Operator:

- <TLDSRSWhoisIPv4>, one or multiple IPv4 addresses to the TLD SRS Whois server
- <TLDSRSWhoisIPv6>, one or multiple IPv6 addresses to the TLD SRS Whois server
- <SRSGwWhoisIPv4>, one or multiple IPv4 addresses to the SRS GW Whois server
- <SRSGwWhoisIPv6>, one or multiple IPv6 addresses to the SRS GW Whois server

2. Test Requirements

2.1 Test items and their identifiers

~~2.1.1 Statement of Work~~

~~The main requirement for testing the Whois service is found in the Statement of Work:~~

> ~~**[R11]** Test the [Registry Operator's] Whois interface for compliance with the requirements described in the Section 5.2 of the AGB, including response format and review of the data mining detection and mitigation control functions.~~

~~Only the first part of the requirement is handled within this test plan. The review of the data mining detection and mitigation control functions are handled as part of the Documentation Test Area.~~

~~2.1.2~~2.1.1 Applicant Guidebook

Section 5.2 of the AGB states the following requirements:

> ***Whois support --*** [Registry Operator] must provision Whois services for the anticipated load. ICANN will verify that Whois data is accessible over IPv4 and IPv6 via both TCP port 43 and via a web interface and review self-certification documentation regarding Whois transaction capacity. Response format according to Specification 4 of the registry agreement and access to Whois (both port 43 and via web) will be tested by ICANN remotely from various points on the Internet over both IPv4 and IPv6.
>
> Self-certification documents shall describe the maximum number of queries per second successfully handled by both the port 43 servers as well as the web interface, together with [a] load expectation [provided by Registry Operator].
>
> Additionally, a description of deployed control functions to detect and mitigate data mining of the Whois database shall be documented.

The following requirements have been identified from the text above. Note that the requirements on Self-certification documents are handled by the Documentation Test Area .

[AGB1] Whois data MUST be accessible over IPv4 via TCP port 43
[AGB2] Whois data MUST be accessible over IPv6 via TCP port 43
[AGB3] Whois data MUST be accessible over IPv4 via a web interface
[AGB4] Whois data MUST be accessible over IPv6 via a web interface

The AGB also states there are no requirements related to IDN.

> Requirements related to IDN for Whois are being developed. After these requirements are developed, prospective registries will be expected to comply with published IDN-related Whois requirements as part of predelegation testing.

~~2.1.3~~2.1.2 Specification 4

Specification 4 of the registry agreement will not be fully cited here, but a number of requirements have been identified. Optional requirements have been removed.

[REG1] Registry Operator MUST operate a WHOIS service available via port 43 in accordance with RFC 3912 at <whois.nic.TLD>
[REG2] Registry Operator MUST operate a web-based Directory Service at <whois.nic.TLD>
[REG3] MUST provide free public query-based access to at least the following elements:
[REG3.1] Domain Name Data
[REG3.2] Registrar Data
[REG3.3] Nameserver Data
[REG8] Offering searchability capabilities on the Directory Services is OPTIONAL but if offered by the Registry Operator it MUST comply with this specification:
[REG8.1] Offer searchability on the web-based Directory Service
[REG8.2] Offer partial match capabilities, at least, on the following fields: domain name, contacts and registrant's name, and contact and registrant's postal address, including all the sub-fields described in EPP (e.g., street, city, state or province, etc.).
[REG8.3] Offer exact-match capabilities, at least, on the following fields: registrar id, name server name, and name server's IP address (only applies to IP addresses stored by the registry, i.e., glue records).
[REG8.4] Offer Boolean search capabilities supporting, at least, the following logical operators to join a set of search criteria: AND, OR, NOT
[REG8.5] Search results will include domain names matching the search criteria.
[REG8.6] Implement appropriate measures to avoid abuse of this feature (e.g., permitting access only to legitimate authorized users)

[REG4-7] were previously replaced by [RDDS-Advisory].

Furthermore, the Registry Agreement states "*ICANN reserves the right to specify alternative formats and protocols, and upon such specification, the Registry Operator will implement such alternative specification as soon as reasonably practicable*". There is a newer alternative to Whois, Registration Data Access Protocol (RDAP). RDAP is not taken into consideration for these tests.

~~2.1.4~~2.1.3 RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy, and Temp-Spec-for-gTLD-data

The 2015-04-27 RDDS-Advisory (see link in reference list) clarifies the interpretation of the Registry Agreement when it comes to valid Whois responses. The Appendix in section 24 of this document contains a detailed specification of a valid Whois response.

[RDDS-Advisory]
The WHOIS service must meet the requirements as specified in the RDDS-Advisory and the appendix in section 24 of this document.

The 2015-08-26 ROID-Advisory (see link in reference list) clarifies the requirements for ROIDs in a valid Whois response.

[ROID-Advisory]
The WHOIS service must meet the requirements as specified in the ROID-Advisory. See the appendix in section 24 of this document.

The 2014-07-02 AWIP (see link in reference list) additionally requires registrars and registries to include in their Whois output information to help Whois users better identify a registration's sponsoring registrar and understand the status codes used by registries and registrars.

[AWIP]
The WHOIS service must meet the requirements as specified in the AWIP. See the appendix in section 24 of this document.

The 2016-07-26 RDDS-Consistent-Labeling-Policy (see link in reference list) changes the requirements on the display of the Whois output information.

[RDDS-Consistent-Labeling-Policy]
The WHOIS service must meet the requirements as specified in the RDDS-Consistent-Labeling-Policy. See the appendix in section 24 of this document.

The 2018-05-17 Temp-Spec-for-gTLD-data (see link in reference list) changes the requirements on the presence of a number of fields in a Domain Name Object. The data type for the email field in contacts in the Domain Name Object has been updated.

[Temp-Spec-for-gTLD-data]
The WHOIS service must meet the requirements as specified in the Temp-Spec-for-gTLD-data. See the appendix in section 24 of this document.

~~2.1.5~~2.1.4 EPP

The requirements state that domain status, individual and organizational names, address, street, city, state/province, postal code, country, telephone and fax numbers, email addresses, date and times MUST conform to the mappings specified in EPP RFCs 5730-5734 and 3915. Below is a listing of the corresponding requirements from those RFCs;

[EPP1] The domain object MUST have at least one of the following status values: clientDeleteProhibited, serverDeleteProhibited, clientHold, serverHold, clientRenewProhibited, serverRenewProhibited, clientTransferProhibited, serverTransferProhibited, clientUpdateProhibited, serverUpdateProhibited, inactive, ok, pendingCreate, pendingDelete, pendingRenew, pendingTransfer, pendingUpdate, addPeriod, autoRenewPeriod, renewPeriod, transferPeriod, redemptionPeriod, pendingRestore, pendingDelete.

[EPP2] Names associated with a contact are represented using character strings.

[EPP3] Contact street, city, and state or province information is represented using character strings.

[EPP4] Contact postal codes are represented using character strings.

[EPP5] Contact country identifiers are represented using two-character identifiers specified in [ISO3166-1].

[EPP6] Telephone numbers described in this mapping are character strings that MUST begin with a plus sign ("+", ASCII value 0x002B), followed by a country code defined in [ITU.E164.2005], followed by a dot (".", ASCII value 0x002E), followed by a sequence of digits representing the telephone number. An optional "x" attribute is provided to note telephone extension information.

[EPP7] Email address syntax is defined in [RFC5322].
[EPP8] Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using upper case "T" and "Z" characters defined in [W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time values.

~~2.1.6~~2.1.5 RFC

Finally, RFC 3912 has a set of requirements on top of those mentioned before:

[RFC1] Each line in the response MUST be ended with ASCII CR and then ASCII LF.
[RFC2] The WHOIS server closes its connection as soon as the output is finished. The closed TCP connection is the indication to the client that the response has been received.

~~2.1.7~~2.1.6 SRS Gateway testing

The Design & Principles document adds a single requirement for SRS Gateway Whois testing.

[SRSGW1] Whois output is fetched both from TLD SRS and SRS Gateway. The output is then compared. They are expected to be equal.

2.2 Features to be tested

The following features of the Whois service will be tested:

- Service on TCP port 43 and HTTP(S) using host <whois.nic.TLD>
- Availability over IPv4 and IPv6
- Queries for known domain names, registrars, and name servers
- Response format as specified in Specification 4 and as interpreted in **RDDS-Advisory**, **ROID-Advisory**, **AWIP**, **RDDS-Consistent-Labeling-Policy** and **Temp-Spec-for-gTLD-data**.
- Searchability is tested if the feature is claimed to be supported
- Search queries based on the requirements in Specification 4

2.3 Features not to be tested

- Values not specified in the specification in the Appendix in section 24 of this document.
- IDN - tested domains must have the ASCII-compatible form (A-Label)
- RDAP
- Data mining protection, as it is part of the Test Area Documentation.

2.4 Approach

The connectivity tests are fully automated. To the extent possible, the test script also validates the response format. Some of the Test Cases do however require manual verification by the Test Officer.

All automated tests are performed over IPv4 and IPv6 from all test nodes, unless explicitly stated otherwise. The IPv4 and IPv6 addresses for the hostname <whois.nic.TLD> will be resolved using the delegation data from the DNS tests, unless explicitly stated otherwise. If the hostname resolves to more than one IPv4 or IPv6 address, all the IP addresses will be tested.

Manual tests, e.g. WhoisWeb03, are generally performed only over IPv4 and from a single node; however, one manual test should be performed over IPv6 from all nodes to verify that the Web Whois service is reachable and responds over IPv6.

The Whois Search test cases are only run if Exhibit A of the Registry Agreement states that the TLD has support for searchable Whois.

2.5 Item pass/fail criteria

The test will pass if an expected response was received from the Whois service. It will however fail if it is not following the requirements.

The Service Level Requirement in Specification 10 of the registry agreement states that "If the RTT is 5 times or more than the corresponding SLR, the RTT will be considered undefined". The requirement for Whois is set to two seconds. A test can thus be failed if it takes longer than 10 seconds to receive an answer from the service.

2.6 Suspension criteria and resumption requirements

The only suspension criteria for the test would be if there were external network problems outside the control of the Registry Operator or the RST tester.

2.7 Test deliverables

The Whois test level will produce:

- Level Test Logs (LTL)
- Anomaly Report (AR) in case of error
- Level Test Report (LTR)

3. Test Traceability Matrix

This table describes the different test cases and their mapping to the requirements.

Test ID	Description	~~Requirement Point~~
WhoisCLI01	Make IPv4 and IPv6 TCP connections on port 43 for all addresses. Query for a known domain name. Verify format of the response.	~~R11, AGB1, REG1, REG3.1, RFC1, RFC2, RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy, Temp-Spec-for-gTLD-data~~
WhoisCLI02	Make IPv4 and IPv6 TCP connections on port 43 for all addresses. Query for a known registrar. Verify format of the response.	~~R11, AGB1, REG1, REG3.2, RFC1, RFC2, RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy~~
WhoisCLI03	Make IPv4 and IPv6 TCP connections on port 43 for all addresses. Query for a known name server. Verify format of the response.	~~R11, AGB1, REG1, REG3.3, RFC1, RFC2, RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy~~
WhoisWeb01	Make an IPv4 HTTP(S) connection to Whois. Verify that there is a successful connection.	~~R11, ABG3, REG2~~
WhoisWeb02	Make an IPv6 HTTP(S) connection to Whois. Verify that there is a successful connection.	~~R11, ABG4, REG2~~
WhoisWeb03	Visit the web-based Whois over IPv4. Query for a known domain name. Verify format of the response.	~~R11, AGB3, REG2, REG3.1, RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy, Temp-Spec-for-gTLD-data~~
WhoisWeb04	Visit the web-based Whois over IPv4. Query for a known registrar. Verify format of the response.	~~R11, AGB3, REG2, REG3.2, RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy~~

Test ID	Description	~~Requirement Point~~
WhoisWeb05	Visit the web-based Whois over IPv4. Query for a known name server. Verify format of the response.	~~R11, AGB3, REG2, REG3.3, RDDS-Advisory, ROID-Advisory, AWIP, RDDS-Consistent-Labeling-Policy~~
WhoisWeb09	Visit the web-based Whois over IPv6. Query for a known domain name. Verify that a Whois response is received.	~~R11, AGB4, REG2, REG3.3~~
WhoisSearch00	Verify support for Searchable Whois. Verify information on how to use Searchable Whois.	~~AGB, Registry Agreement~~
WhoisSearch01	Visit the web-based Whois over IPv4. Verify abuse protection.	~~R11, AGB3, REG2, REG8.6~~
WhoisSearch02	Visit the web-based Whois over IPv4. Perform partial match queries. Verify that the response contains the expected result.	~~R11, AGB3, REG2, REG8.1, REG8.2, REG8.5, Temp-Spec-for-gTLD-data~~
WhoisSearch03	Visit the web-based Whois over IPv4. Perform exact-match queries. Verify that the response contains the expected result.	~~R11, AGB3, REG2, REG8.1, REG8.3, REG8.5~~
WhoisSearch04	Visit the web-based Whois over IPv4. Perform boolean queries. Verify that the response contains the expected result.	~~R11, AGB3, REG2, REG8.1, REG8.4, REG8.5~~
WhoisSearch09	Visit the web-based Whois over IPv6. Perform a search query. Verify that a response to the search is received.	~~R11, AGB4, REG2, REG8.1~~
SRS+GW Whois CLI 01	Verify that a domain name object is synchronized between the Gateway Registry System and the TLD Registry System.	~~SRSGW1, Temp-Spec-for-gTLD-data~~
SRS+GW Whois CLI 02	Verify that a registrar object is synchronized between the Gateway Registry System and the TLD Registry System.	~~SRSGW1~~
SRS+GW Whois CLI 03	Verify that a name server object is synchronized between the Gateway Registry System and the TLD Registry System.	~~SRSGW1~~

4. Test management

The goal of these documents is to describe the test cases and how the new gTLDs are tested. This is just a part of a larger project and defining test management is not part of this subproject. However, some information can be found in the Master Test Plan.

5. Test Case WhoisCLI01: Query for Domain Name

5.1 Test case identifier

WhoisCLI01

5.2 Objective

Multiple Domain Name Object queries will be performed over IPv4 and IPv6 TCP connections on port 43. The objective is to verify the format of the responses of such queries.

5.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestDomain	An existing domain name which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String
CharEncoding	Character encoding declaration	String
EppRepID	EPP Repository Identifier declaration	String

5.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", in this document and all responses must be identical.

5.5 Environmental needs

- Whois client software that preserves line endings in the response, e.g. Netcat
- IPv4 connectivity
- IPv6 connectivity
- List of IP addresses (IPv4 and IPv6, respectively) from resolving <whois.nic.TLD> using the delegation data, where <WhoisIP> is one such address.
- <CharEncoding>, character encoding declaration from pdtwhois.xml file.
- <EppRepID>, EPP Repository Identifier declaration from pdtwhois.xml file

5.6 Special procedural requirements

A Whois query and its response MUST NOT take longer than 10 seconds.

5.7 Intercase dependencies

This test has no intercase dependencies.

5.8 Ordered description of steps to be taken to execute the test case

1. Resolve IPv4 and IPv6 addresses of the host whois.nic.**<TLD>** using **<DnsGlueRecord>**. If multiple addresses are returned from lookup, all addresses are used.
2. Make a query from each test node using the client software, i.e. the same query is sent from each test node and to all IP addresses found (IPv4 and IPv6).
   whois -h **<WhoisIP> <WhoisTestDomain>**

3. For each query executed repeat the following steps:
    a. Verify that the Whois server returns a response.
    b. Verify that the character encoding of the response matches <CharEncoding>.
    c. Verify that the response is identical to all other responses.
    d. Verify that the response is a "Domain name reply" as defined in section 24.
    e. Verify that the response conforms to the format in the specification. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section* 24 *in this document for all details.*
    f. Verify that the queried domain name exactly matches the name in the "Domain Name" field in the response.
    g. Verify that the *ROID Suffix*, as defined in section 24, of the "Domain ID" field exactly matches <EppRepID>.

Criteria for PASS:

- The Whois output is encoded in ASCII, and
- The Whois service is available on IPv4 and IPv6, and
- All verifications (steps 3a-g) are successful.

Criteria for WARN:

- The Whois output is not encoded in ASCII, and
- The Whois output was successfully converted to UTF-8, and
- All other criteria for PASS are successful.

Criteria for FAIL:

- The encoding of the Whois output cannot be converted to UTF-8, or
- At least one other criteria for PASS is unsuccessful.

6. Test Case WhoisCLI02: Query for Registrar

6.1 Test case identifier

WhoisCLI02

6.2 Objective

Multiple Registrar Object queries will be performed over IPv4 and IPv6 TCP connections on port 43. The objective is to verify the format of the responses of such queries.

6.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestRegistrar	An existing registrar which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String
CharEncoding	Character encoding of Whois output	String

6.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", in this document and all responses must be identical.

6.5 Environmental needs

- Whois client software that preserves line endings in the response, e.g. Netcat
- IPv4 connectivity
- IPv6 connectivity
- List of IP addresses (IPv4 and IPv6, respectively) from resolving <whois.nic.TLD> using the delegation data, where <WhoisIP> is one such address.
- <CharEncoding>, character encoding declaration from pdtwhois.xml file.

6.6 Special procedural requirements

A Whois query and its response MUST NOT take longer time than 10 seconds.

6.7 Intercase dependencies

This test has no intercase dependencies.

6.8 Ordered description of steps to be taken to execute the test case

1. Resolve IPv4 and IPv6 addresses of the host whois.nic.<TLD> using <DnsGlueRecord>. If multiple addresses are returned from lookup, all addresses are used.
2. Make a query from each test node using the client software, i.e. the same query is sent from each test node and to all IP addresses found (IPv4 and IPv6).
   whois -h **<WhoisIP> < WhoisTestRegistrar >**
3. For each query executed repeat the following steps:
   a. Verify that the Whois server returns a response.

b. Verify that the character encoding of the response matches <CharEncoding>.
c. Verify that the response is identical to all other responses.
d. Verify that the response is a "Registrar reply" as defined in section 24.
e. Verify that the response conforms to the format in the specification. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section* 24 *in this document for all details.*
f. Verify that the queried registrar name matches the name in all "Registrar Name" fields in the response. The match may be a substring match.

Criteria for PASS:

- The Whois output is encoded in ASCII, and
- The Whois service is available on IPv4 and IPv6, and
- All verifications (steps 3a-f) are successful.

Criteria for WARN:

- The Whois output is not encoded in ASCII, and
- The Whois output was successfully converted to UTF-8, and
- All other criteria for PASS are successful.

Criteria for FAIL:

- The encoding of the Whois output cannot be converted to UTF-8, or
- At least one other criteria for PASS is unsuccessful.

7. Test Case WhoisCLI03: Query for Name Server

7.1 Test case identifier

WhoisCLI03

7.2 Objective

Multiple Name Server Object queries will be performed over IPv4 and IPv6 TCP connections on port 43. The objective is to verify the format of the responses of such queries.

7.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestNameServerName	The domain name of an existing name server which has Whois data	String
WhoisTestNameServerIP	The IP address of an existing name server which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String
CharEncoding	Character encoding of Whois output	String

7.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", in this document and all responses must be identical.

7.5 Environmental needs

- Whois client software that preserves line endings in the response, e.g. Netcat
- IPv4 connectivity
- IPv6 connectivity
- List of IP addresses (IPv4 and IPv6, respectively) from resolving <whois.nic.TLD> using the delegation data, where <WhoisIP> is one such address.
- <CharEncoding>, character encoding declaration from pdtwhois.xml file.

7.6 Special procedural requirements

Abort the test if any Whois query takes longer than 10 seconds.

7.7 Intercase dependencies

This test has no intercase dependencies.

7.8 Ordered description of steps to be taken to execute the test case

1. Resolve IPv4 and IPv6 addresses of the host whois.nic.<TLD> using <DnsGlueRecord>. If multiple addresses are returned from lookup, all addresses are used.
2. Make two queries from each test node using the client software, i.e. the same two queries are sent from each test node and to all IP addresses found (IPv4 and IPv6).
   whois -h **<WhoisIP>** "nameserver **<WhoisTestNameServerName>**"
   whois -h **<WhoisIP>** "nameserver **<WhoisTestNameServerIP>**"
3. Verify that the Whois server returns responses to the two queries. A response means at least some text is returned.
4. Both queries (WhoisTestNameServerName or WhoisTestNameServerIP) for all queried Whois servers MUST give a response that matches the following steps and requirements.
   a. Verify that the character encoding of the responses match <CharEncoding>.
   b. Verify that the response is identical to all other responses to the same query.
   c. Verify that the response is a "Name server reply type 1" or "Name server reply type 2" as defined in section 24.
   d. Verify that response conforms to the format in the specification. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section* 24 *in this document for all details.*
   e. If the response is "Name server reply type 1" and the query is for WhoisTestNameServerName do the following step:
      i. Verify that the "Server Name" field field exactly matches WhoisTestNameServerName in all returned Name Server objects.
   f. If the response is "Name server reply type 1" and the query is for WhoisTestNameServerIP do the following step:
      i. If the "IP Address" field is present, verify that at least one such field exactly matches WhoisTestNameServerIP.
      ii. Repeat the previous verification in every Name Server objects in the response.
   g. If the response is "Name server reply type 2" do the following step:
      i. Extract the ROID from the first "Roid Line" as defined in the format specification.
      ii. Make a new whois query, now using the extracted ROID in the query string ("roid <ROID>").
      iii. Verify that the new response is a "Name server reply type 1" as defined in section 24.
      iv. Verify that the new response conforms to the format in the specification.
      v. Verify the response using step e or f above.

Criteria for PASS:

- The Whois outputs are encoded in ASCII, and
- The Whois service is available on IPv4 and IPv6, and
- All verifications (under steps 3 and 4a-g) are successful.

Criteria for WARN:

- The Whois outputs are not encoded in ASCII, and
- The Whois output was successfully converted to UTF-8, and
- All other criteria for PASS are successful.

Criteria for FAIL:

- The encoding of the Whois output cannot be converted to UTF-8, or

- At least one other criteria for PASS is unsuccessful.

8. Test Case WhoisWeb01: Verify IPv4 Connectivity

8.1 Test case identifier

WhoisWeb01

8.2 Objective

The automated test makes HTTP and HTTPS connections over IPv4 from all nodes and verifies that there is a successful connection. The test does not validate the certificate for the HTTPS connection.

8.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String

8.4 Outcome(s)

The Whois service MUST be available on HTTP over IPv4.

8.5 Environmental needs

- Whois test script
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

8.6 Special procedural requirements

Connect time must not exceed 10 seconds.

8.7 Intercase dependencies

This test has no intercase dependencies.

8.8 Ordered description of steps to be taken to execute the test case

1. Resolve the IPv4 addresses of the host whois.nic.**<TLD>** using **<DnsGlueRecord>**. If multiple addresses are returned from lookup, all addresses are used. <WhoisIPv4> is one such address
2. Repeat the following steps from each test node for every IPv4 address:
    a. Make an IPv4 HTTP connection to whois.nic.**<TLD>** (**<WhoisIPv4>**). Redirection is allowed but the end HTTP response MUST be 200.
    b. Make an IPv4 HTTPS connection to whois.nic.**<TLD>** (**<WhoisIPv4>**). Redirection is allowed but the end HTTP response SHOULD be 200.
    c. If the HTTP connection returns a referral to HTTPS, then that referral and any subsequent referrals MUST return an HTTP status response 200.
3. To pass the test, every IPv4 address must pass.

Criteria for PASS:

- The web HTTP Whois page and the web HTTPS Whois page can be reached using IPv4 only, and

- the HTTP status code is 200 for both, and
- the criteria are met from all test nodes, and
- the criteria are met for every IPv4 address.

Criteria for WARN:

- The web HTTP Whois page can be reached using IPv4 only, and
- the HTTP status code is 200 for the HTTP page, and
- the web HTTPS Whois page cannot be reached or cannot be reached using IPv4 only, or
- the HTTP status code is not 200 for the HTTPS page.
- Failure of HTTPS on one IPv4 address is enough to get WARN.

Criteria for FAIL:

- The web HTTP Whois page cannot be reached or cannot be reached using IPv4 only, or
- the HTTP status code is not 200 for the HTTP page.
- Failure of HTTP on one IPv4 address is enough to get FAIL.

9. Test Case WhoisWeb02: Verify IPv6 Connectivity

9.1 Test case identifier

WhoisWeb02

9.2 Objective

The automated test makes HTTP and HTTPS connections over IPv6 from all nodes and verifies that there is a successful connection. The test does not validate the certificate for the HTTPS connection.

9.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String

9.4 Outcome(s)

The Whois service MUST be available on HTTP over IPv6.

9.5 Environmental needs

- Whois test script
- IPv6 connectivity
- <WhoisIPv6> from resolving <whois.nic.TLD> using the delegation data.

9.6 Special procedural requirements

Connect time must not exceed 10 seconds.

9.7 Intercase dependencies

This test has no intercase dependencies.

9.8 Ordered description of steps to be taken to execute the test case

1. Resolve the IPv6 addresses of the host whois.nic.<TLD> using <DnsGlueRecord>. If multiple addresses are returned from lookup, all addresses are used. <WhoisIPv6> is one such address
2. Repeat the following steps from each test node for every IPv6 address:
    a. Make an IPv6 HTTP connection to whois.nic.**<TLD>** (**<WhoisIPv6>**). Redirection is allowed but the end HTTP response MUST be 200.
    b. Make an IPv6 HTTPS connection to whois.nic.**<TLD>** (**<WhoisIPv6>**). Redirection is allowed but the end HTTP response SHOULD be 200.
    c. If the HTTP connection returns a referral to HTTPS, then that referral and any subsequent referrals MUST return an HTTP status response 200.
3. To pass the test, every IPv6 address must pass.

Criteria for PASS:

- The web HTTP Whois page and the web HTTPS Whois page can be reached using IPv6 only, and

- the HTTP status code is 200 for both, and
- the criteria are met from all test nodes, and
- the criteria are met for every IPv6 address.

Criteria for WARN:

- The web HTTP Whois page can be reached using IPv6 only, and
- the HTTP status code is 200 for the HTTP page, and
- the web HTTPS Whois page cannot be reached or cannot be reached using IPv6 only, or
- the HTTP status code is not 200 for the HTTPS page.
- Failure of HTTPS on one IPv6 address is enough to get WARN.

Criteria for FAIL:

- The web HTTP Whois page cannot be reached or cannot be reached using IPv6 only, or
- the HTTP status code is not 200 for the HTTP page.
- Failure of HTTP on one IPv6 address is enough to get FAIL.

10. Test Case WhoisWeb03: Manual Query for Domain Name

10.1 Test case identifier

WhoisWeb03

10.2 Objective

A manual query for a known domain name will be performed on the Whois IPv4 website. The objective is to verify the format of the response.

10.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestDomain	An existing domain name which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String
EppRepID	EPP Repository Identifier declaration	String

10.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", in this document and the communication MUST be over IPv4.

10.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.
- <EppRepID>, EPP Repository Identifier declaration from pdtwhois.xml file

10.6 Special procedural requirements

Lookup time must not exceed 10 seconds.

10.7 Intercase dependencies

This test has no intercase dependencies.

10.8 Ordered description of steps to be taken to execute the test case

1. Resolve the IPv4 addresses of the host whois.nic.**<TLD>** using **<DnsGlueRecord>**. If multiple addresses are returned from lookup, one address is used.
2. Browse to http://whois.nic.**<TLD>**.
3. Make a query for the domain **<WhoisTestDomain>**.
4. Copy the Whois response from the web page and paste it into a text file.
5. Verify that the Whois response matches the format specification given in section 24. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section* 24 *in this document for all details.*
6. Verify that the response is a "Domain name reply" as defined in the format specification.

7. Verify that the queried domain name exactly matches the name in the "Domain Name" field in the response.
8. Verify that the *ROID Suffix*, as defined in specification in the Whois TP, of the "Domain ID" field exactly matches <EppRepID>.

Criteria for PASS:

- All verifications (steps 5-8) are successful.

Criteria for FAIL:

- At least one criteria for PASS is unsuccessful.

11. Test Case WhoisWeb04: Manual Query for Registrar

11.1 Test case identifier

WhoisWeb04

11.2 Objective

A manual query for a known registrar will be performed on the Whois IPv4 website. The objective is to verify the format of the response.

11.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestRegistrar	An existing registrar which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String

11.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", in this document and the communication MUST be over IPv4.

11.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

11.6 Special procedural requirements

Lookup time must not exceed 10 seconds.

11.7 Intercase dependencies

This test has no intercase dependencies.

11.8 Ordered description of steps to be taken to execute the test case

1. Resolve the IPv4 addresses of the host whois.nic.<TLD> using <DnsGlueRecord>. If multiple addresses are returned from lookup, one address is used.
2. Browse to http://whois.nic.**<TLD>**.
3. Make a query for the registrar **<WhoisTestRegistrar>**.
4. Copy the Whois response from the web page and paste it into a text file.
5. Verify that the Whois response matches the format specification in section 24. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section 24 in this document for all details.*
6. Verify that the response is a "Registrar reply" as defined in the format specification.
7. Verify that the queried registrar name matches the name in all "Registrar Name" fields in the response. The match may be a substring match.

Criteria for PASS:

- All verifications (steps 5-7) are successful.

Criteria for FAIL:

- At least one criteria for PASS is unsuccessful.

12. Test Case WhoisWeb05: Manual Query for Name Server

12.1 Test case identifier

WhoisWeb05

12.2 Objective

A manual query for a known name server will be performed on the Whois IPv4 website. The objective is to verify the format of the response.

12.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestNameServerName	The domain name of an existing name server which has Whois data	String
WhoisTestNameServerIP	The IP address of an existing name server which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String

12.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", in this document and the communication MUST be over IPv4.

12.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

12.6 Special procedural requirements

Lookup time must not exceed 10 seconds.

12.7 Intercase dependencies

This test has no intercase dependencies.

12.8 Ordered description of steps to be taken to execute the test case

1. Resolve the IPv4 addresses of the host whois.nic.<TLD> using <DnsGlueRecord>. If multiple addresses are returned from lookup, one address is used.
2. Browse to http://whois.nic.**<TLD>**.
3. Make a query for the name server using the hostname **<WhoisTestNameServerName>** and the IP **<WhoisTestNameServerIP>**, respectively. At least one of them MUST give a response with Whois data.
4. Verify that both queries give some response.
5. Copy the Whois response from the web page and paste it into a text file.

6. Verify that the Whois response matches the format specification given in section 24. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section 24 in this document for all details.*
7. Verify that the response is a "Name server reply type 1" or "Name server reply type 2" as defined in the format specification.
8. If the response is "Name server reply type 1" do the following step:
    a. Verify that the "Server Name" field or the "IP Address" field exactly matches the query string from step 3 above.
9. If the response is "Name server reply type 2" do the following step:
    a. Extract the ROID from the first "Roid Line" as defined in the format specification.
    b. Make a new whois query, now using the extracted ROID as the query string.
    c. Copy the Whois response from the web page and paste it into a text file.
    d. Verify that the new response is a "Name server reply type 1" as defined in the format specification.
    e. Verify that the "Server Name" field or the "IP Address" field in the new response exactly matches the query string from step 3 above.

Criteria for PASS:

- All verifications (steps 4 and 6-9, as applicable) are successful.

Criteria for FAIL:

- At least one criteria for PASS is unsuccessful.

13. Whois Web 09: Manual Query over IPv6

13.1 Test case identifier

WhoisWeb09

13.2 Objective

A manual query for a known domain name will be performed on the Whois IPv6 website. The objective is to verify that the Whois service is reachable and responds over IPv6.

13.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestDomain	An existing domain name which has Whois data	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String

13.4 Outcome(s)

The Whois service MUST be reachable over IPv6 and it MUST return a valid response.

13.5 Environmental needs

- Web browser
- IPv6 connectivity
- <WhoisIPv6> from resolving <whois.nic.TLD> using the delegation data.

13.6 Special procedural requirements

Lookup time must not exceed 10 seconds.

13.7 Intercase dependencies

This test has no intercase dependencies.

13.8 Ordered description of steps to be taken to execute the test case

1. Resolve the IPv6 addresses of the host whois.nic.<TLD> using <DnsGlueRecord>. If multiple addresses are returned from lookup, one address is used.
2. Browse to http://whois.nic.**<TLD>**.
3. Verify that all communication is done over IPv6 and that no intermediate step is done over IPv4.
4. Make a query for the domain **<WhoisTestDomain>**.
5. The queried domain name MUST be present in the response. Beyond that, the response content and format will not be verified.

Criteria for PASS:

- All verifications (steps 3 and 5) are successful.

Criteria for FAIL:

- At least one verification (step 3 or 5) is unsuccessful.

14. Test Case WhoisSearch00: Verify if Whois Search is Supported

14.1 Test case identifier

WhoisSearch00

14.2 Objective

The objective is to verify if the Registry Operator is contractually obligated to support searchable Whois service. If the RO will support searchable Whois, instructions must be provided on how to use the searchable Whois service.

14.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
RegAgree	The Registry Agreement for the specific gTLD	Document
SearchWhoisDoc	URL to a manual for searchable whois for the specific gTLD	String

14.4 Outcome(s)

- If searchable Whois is required by the Registry Agreement for the specific gTLD then remainder of the test cases must be performed, and the Registry Operator must provide instructions on how to use searchable Whois service.
- If the Registry Agreement does not require support for searchable Whois no further tests will be performed.

14.5 Environmental needs

- Web browser

14.6 Special procedural requirements

Lookup time must not exceed 10 seconds.

14.7 Ordered description of steps to be taken to execute the test case

1. Identify support for searchable Whois within Registry Agreement. Expected location is Exhibit A. Determine if support for searchable Whois is required by contract.
   If not, the test will end and the result of this test case and the following test cases are N/A.
2. If it exists continue:
   a. Browse to the web page where information on how to use searchable Whois is located.
   b. Information must be presented in English.
   c. Make a note on any visible information regarding how to use searchable Whois.
   d. If the solution for searching is self-explanatory then no separate information is needed.

Criteria for N/A:

- Searchable Whois is not required by Exhibit A.

Criteria for PASS:

- Searchable Whois is required by Exhibit A, and
- enough information on how to use searchable Whois is available, and
- the information is in English, and
- the information is comprehensible and complete.

Criteria for FAIL:

- Searchable Whois is required by Exhibit A but is not available, or
- no information on how to use searchable Whois is available, or
- the information is not in English, or
- the information is incomplete or not comprehensible.

15. Test Case WhoisSearch01: Verify Abuse Protection

15.1 Test case identifier

WhoisSearch01

15.2 Objective

The objective is to verify the abuse protection.

15.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisCredentials	Optional credentials. E.g. username and password for accessing the Whois service	String
DnsGlueRecord	IPv4 or IPv6 addresses for auth NS	String

15.4 Outcome(s)

Abuse protection MUST exist for the Whois service.

15.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

15.6 Special procedural requirements

Whois query response time must not exceed 10 seconds.

15.7 Intercase dependencies

This test has no intercase dependencies.

15.8 Ordered description of steps to be taken to execute the test case

1. Resolve IP addresses of the host whois.nic.**<TLD>** using **<DnsGlueRecord>**. This result will also be used for subsequent tests.
2. Make sure that whois.nic.**<TLD>** resolves to **<WhoisIPv4>**.
3. Browse to http://whois.nic.**<TLD>**
4. Make a note on any visible abuse protection.
5. Access the service with **<WhoisCredentials>**, if required.

Criteria for N/A:

- Searchable Whois is not required by Exhibit A.

Criteria for PASS:

- Abuse protection is used.

Criteria for FAIL:

- No abuse protection can be detected.

16. Test Case WhoisSearch02: Partial Match Capabilities

16.1 Test case identifier

WhoisSearch02

16.2 Objective

Manual search queries will be performed on the Whois IPv4 website. The objective is to verify the partial match capabilities.

16.3 Inputs

- Information on webpage in English on how to use the Searchable Whois.
- The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisCredentials	Optional credentials. E.g. username and password for accessing the Whois service	String
WhoisTestDomain	An existing domain which has Whois data	String

16.4 Outcome(s)

The Whois service MUST offer partial match capabilities.

16.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

16.6 Special procedural requirements

Whois query response time must not exceed 10 seconds.

16.7 Intercase dependencies

This test has no intercase dependencies.

16.8 Ordered description of steps to be taken to execute the test case

1. Make sure that whois.nic.**<TLD>** resolves to **<WhoisIPv4>**.
2. Browse to http://whois.nic.**<TLD>**
3. Access the service with **<WhoisCredentials>**, if required.
4. For the following tests, use data from previous domain query. If the data field or fields have been redacted based on **Temp-Spec-for-gTLD-data**, the test is skipped.
    a. Perform a partial match search based on domain name.
    b. Perform a partial match search based on contact name unless redacted.
    c. Perform a partial match search based on registrant name unless redacted.
    d. Perform a partial match search based on contact postal address (e.g. street, city, state, or province) unless redacted.

e. Perform a partial match search based on registrant postal address (e.g. street, city, state, or province) unless redacted.

5. The search results MUST include the domain name(s) matching the search criteria.

Criteria for N/A:

- Searchable Whois is not required by Exhibit A.

Criteria for PASS:

- A partial match search based on domain name was successfully completed, and
- a partial match search based on contact name was successfully completed or skipped, and
- a partial match search based on registrant name was successfully completed or skipped, and
- a partial match search based on contact postal address was successfully completed or skipped, and
- a partial match search based on registrant postal address was successfully completed or skipped, and
- all search results included the domain name(s) matching the search criteria.

Criteria for FAIL:

- Any search query failed to complete, or
- Any search result failed to include the domain name(s) matching the search criteria.

17. Test Case WhoisSearch03: Exact Match Search

17.1 Test case identifier

WhoisSearch03

17.2 Objective

Manual search queries will be performed on the Whois IPv4 website. The objective is to verify the exact-match capabilities.

17.3 Inputs

- Information on webpage in English on how to use the Searchable Whois.
- The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisCredentials	Optional credentials. E.g. username and password for accessing the Whois service	String
WhoisTestDomain	An existing domain which has Whois data	String

17.4 Outcome(s)

The Whois service MUST offer exact-match capabilities.

17.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

17.6 Special procedural requirements

Whois query response time must not exceed 10 seconds.

17.7 Intercase dependencies

This test has no intercase dependencies.

17.8 Ordered description of steps to be taken to execute the test case

1. Make sure that whois.nic.**<TLD>** resolves to **<WhoisIPv4>**.
2. Browse to http://whois.nic.**<TLD>**
3. Access the service with **<WhoisCredentials>**, if required.
4. Perform an exact-match search based on registrar id. Use data from a previous domain query.
5. Perform an exact-match search based on name server name. Use data from a previous domain query.
6. The search results MUST include the domain name(s) matching the search criteria.

Criteria for N/A:

- Searchable Whois is not required by Exhibit A.

Criteria for PASS:

- An exact match search based on registrar id was successfully completed, and
- an exact match search based on name server name was successfully completed, and
- all search results included the domain name(s) matching the search criteria.

Criteria for FAIL:

- Any search query failed to complete, or
- Any search result failed to include the domain name(s) matching the search criteria.

18. Test Case WhoisSearch04: Boolean Search Capabilities

18.1 Test case identifier

WhoisSearch04

18.2 Objective

Manual search queries will be performed on the Whois IPv4 website. The objective is to verify the Boolean search capabilities.

18.3 Inputs

- Information on webpage in English on how to use the Searchable Whois.
- The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisCredentials	Optional credentials. E.g. username and password for accessing the Whois service	String
WhoisTestDomain	An existing domain name which has Whois data	String

18.4 Outcome(s)

The Whois service MUST offer Boolean search capabilities.

18.5 Environmental needs

- Web browser
- IPv4 connectivity
- <WhoisIPv4> from resolving <whois.nic.TLD> using the delegation data.

18.6 Special procedural requirements

Whois query response time must not exceed 10 seconds.

18.7 Intercase dependencies

This test has no intercase dependencies.

18.8 Ordered description of steps to be taken to execute the test case

1. Make sure that whois.nic.**<TLD>** resolves to **<WhoisIPv4>**.
2. Browse to http://whois.nic.**<TLD>**
3. Access the service with **<WhoisCredentials>**, if required.
4. Perform a Boolean search using AND based on data from a previous domain query.
5. Perform a Boolean search using OR based on data from a previous domain query.
6. Perform a Boolean search using NOT based on data from a previous domain query.
7. The search results MUST include the domain name(s) matching the search criteria.

Criteria for N/A:

- Searchable Whois is not required by Exhibit A.

Criteria for PASS:

- A Boolean search using AND was successfully completed, and
- a Boolean search using OR was successfully completed, and
- a Boolean search using NOT was successfully completed, and
- all search results included the domain name(s) matching the search criteria.

Criteria for FAIL:

- Any search query failed to complete, or
- Any search result failed to include the domain name(s) matching the search criteria.

19. Test Case WhoisSearch09: Search over IPv6

19.1 Test Case Identifier

WhoisSearch09

19.2 Objective

A manual search query will be performed on the Whois IPv6 website. The objective is to verify that the search is reachable and responds over IPv6.

19.3 Inputs

- Information on webpage in English on how to use the Searchable Whois.
- The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisCredentials	Optional credentials. E.g. username and password for accessing the Whois service	String
WhoisTestDomain	An existing domain name which has Whois data	String

19.4 Outcome(s)

The Whois service MUST be reachable over IPv6 and it MUST return a valid response.

19.5 Environmental needs

- Web browser
- IPv6 connectivity
- <WhoisIPv6> from resolving <whois.nic.TLD> using the delegation data.

19.6 Special procedure requirements

Whois query response time must not exceed 10 seconds.

19.7 Intercase dependencies

This test has no intercase dependencies.

19.8 Ordered description of steps to be taken to execute the test case

1. Make sure that whois.nic.**<TLD>** resolves to **<WhoisIPv6>**.
2. Browse to http://whois.nic.**<TLD>**
3. Access the service with **<WhoisCredentials>**, if required.
4. Verify that all communication is done over IPv6 and that no intermediate step is done over IPv4.
5. Perform a search. Use the data from a previous domain query.
6. The search result MUST be successful.

Criteria for N/A:

- Searchable Whois is not required by Exhibit A.

Criteria for PASS:

- All communication is done over IPv6, and
- the search was successfully completed.

Criteria for FAIL:

- Some communication is done over IPv4, or
- the search did not successfully complete, or
- the search result did not include the domain name matching the search criteria.

20. Test Case SRS+GW Whois CLI 01: Verify consistency for Domain Name Objects

20.1 Test case identifier

SRSGWWhoisCLI01

20.2 Objective

Multiple Domain Name Object queries will be performed over IPv4 and IPv6 TCP connections on port 43 (see section 1.6 for information on the availability of IPv6 connectivity in the local area test nodes). The objective is to verify the format of the responses of such queries and to verify that the TLD SRS Whois server gives the same response as the SRS GW Whois server.

20.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestDomain	An existing domain name which has Whois data	String
CharEncoding	Character encoding declaration	String
EppRepID	EPP Repository Identifier declaration	String
TLDSRSWhoisIPv4	The IPv4 address or addresses to the TLD SRS Whois server	String
TLDSRSWhoisIPv6	The IPv6 address or addresses to the TLD SRS Whois server	String
SRSGwWhoisIPv4	The IPv4 address or addresses to the SRS GW Whois server	String
SRSGwWhoisIPv6	The IPv6 address or addresses to the SRS GW Whois server	String

20.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", and all responses must be identical.

20.5 Environmental needs

- Whois client software that preserves line endings in the response, e.g. Netcat
- IPv4 connectivity
- IPv6 connectivity (if available)
- List of IP addresses (IPv4 and IPv6, respectively) from resolving <whois.nic.TLD> using the delegation data, where <WhoisIP> is one such address.
- <CharEncoding>, character encoding declaration from pdtwhois.xml file.
- <EppRepID>, EPP Repository Identifier declaration from pdtwhois.xml file
- Test node outside the country of the SRS GW <TestNodeA> to be used to test TLD SRS Whois server from.
- Test node inside the country of the SRS GW <TestNodeB> to be used to test SRS GW Whois server from.

20.6 Special procedural requirements

A Whois query and its response MUST NOT take longer than 10 seconds.

20.7 Intercase dependencies

This test has no intercase dependencies.

20.8 Ordered description of steps to be taken to execute the test case

1. Make queries from TestNodeA to the TLD SRS Whois server using the client software, i.e. to all IP addresses found.
   whois -h **<WhoisIP> <WhoisTestDomain>**
2. Make queries from TestNodeB to the SRS GW Whois server using the client software, i.e. to all IP addresses submitted.
   whois -h **<WhoisIP> <WhoisTestDomain>**
3. For each query executed repeat the following steps:
   a. Verify that the Whois server returns a response.
   b. Verify that the character encoding of the response matches <CharEncoding>.
   c. Verify that the response is identical to all other responses.
   d. Verify that the response is a "Domain name reply" as defined in the Whois TP.
   e. Verify that the response conforms to the format in the specification in the Whois TP. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section 24 for all details.*
   f. Verify that the queried domain name exactly matches the name in the "Domain Name" field in the response.
   g. Verify that the *ROID Suffix*, as defined in specification, of the "Domain ID" field exactly matches <EppRepID>.

Criteria for PASS:

- The Whois output is encoded in ASCII, and
- The Whois service is available on IPv4 (always) and IPv6 (if IPv6 connectivity is available in the local area test node), and
- All verifications (steps 3a-g) are successful.

Criteria for WARN:

- The Whois output is not encoded in ASCII, and
- The Whois output was successfully converted to UTF-8, and
- All other criteria for PASS are successful.

Criteria for FAIL:

- The encoding of the Whois output cannot be converted to UTF-8, or
- At least one other criteria for PASS is unsuccessful.

21. Test Case SRS+GW Whois CLI 02: Verify consistency for Registrar objects

21.1 Test case identifier

SRSGWWhoisCLI02

21.2 Objective

Multiple Registrar Object queries will be performed over IPv4 and IPv6 TCP connections on port 43 (see section 1.6 for information on the availability of IPv6 connectivity in the local area test nodes). The objective is to verify the format of the responses of such queries and to verify that the TLD SRS Whois server gives the same response as the SRS GW Whois server.

21.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestRegistrar	An existing registrar which has Whois data	String
CharEncoding	Character encoding of Whois output	String
TLDSRSWhoisIPv4	The IPv4 address or addresses to the TLD SRS Whois server	String
TLDSRSWhoisIPv6	The IPv6 address or addresses to the TLD SRS Whois server	String
SRSGwWhoisIPv4	The IPv4 address or addresses to the SRS GW Whois server	String
SRSGwWhoisIPv6	The IPv6 address or addresses to the SRS GW Whois server	String

21.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", and all responses must be identical.

21.5 Environmental needs

- Whois client software that preserves line endings in the response, e.g. Netcat
- IPv4 connectivity
- IPv6 connectivity (if available)
- List of IP addresses (IPv4 and IPv6, respectively) from resolving <whois.nic.TLD> using the delegation data, where <WhoisIP> is one such address.
- <CharEncoding>, character encoding declaration from pdtwhois.xml file.
- Test node outside the country of the SRS GW <TestNodeA> to be used to test TLD SRS Whois server from.
- Test node inside the country of the SRS GW <TestNodeB> to be used to test SRS GW Whois server from.

21.6 Special procedural requirements

A Whois query and its response MUST NOT take longer time than 10 seconds.

21.7 Intercase dependencies

This test has no intercase dependencies.

21.8 Ordered description of steps to be taken to execute the test case

1. Make queries from TestNodeA to the TLD SRS Whois server using the client software, i.e. to all IP addresses found.
   whois -h **<WhoisIP> < WhoisTestRegistrar >**
2. Make queries from TestNodeB to the SRS GW Whois server using the client software, i.e. to all IP addresses submitted.
   whois -h **<WhoisIP> < WhoisTestRegistrar >**
3. For each query executed repeat the following steps:
   a. Verify that the Whois server returns a response.
   b. Verify that the character encoding of the response matches <CharEncoding>.
   c. Verify that the response is identical to all other responses.
   d. Verify that the response is a "Registrar reply " as defined in the Whois TP.
   e. Verify that the response conforms to the format in the specification. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section 24 for all details.*
   f. Verify that the queried registrar name matches the name in all "Registrar Name" fields in the response. The match may be a substring match.

Criteria for PASS:

- The Whois output is encoded in ASCII, and
- The Whois service is available on IPv4 (always) and IPv6 (if IPv6 connectivity is available in the local area test node), and
- All verifications (steps 3a-f) are successful.

Criteria for WARN:

- The Whois output is not encoded in ASCII, and
- The Whois output was successfully converted to UTF-8, and
- All other criteria for PASS are successful.

Criteria for FAIL:

- The encoding of the Whois output cannot be converted to UTF-8, or
- At least one other criteria for PASS is unsuccessful.

22. Test Case SRS+GW Whois CLI 03: Verify consistency for Name Server objects

22.1 Test case identifier

SRSGWWhoisCLI03

22.2 Objective

Multiple Name Server Object queries will be performed over IPv4 and IPv6 TCP connections on port 43 (see section 1.6 for information on the availability of IPv6 connectivity in the local area test nodes). The objective is to verify the format of the responses of such queries and to verify that the TLD SRS Whois server gives the same response as the SRS GW Whois server.

22.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
WhoisTestNameServerName	The domain name of an existing name server which has Whois data	String
WhoisTestNameServerIP	The IP address of an existing name server which has Whois data	String
CharEncoding	Character encoding of Whois output	String
TLDSRSWhoisIPv4	The IPv4 address or addresses to the TLD SRS Whois server	String
TLDSRSWhoisIPv6	The IPv6 address or addresses to the TLD SRS Whois server	String
SRSGwWhoisIPv4	The IPv4 address or addresses to the SRS GW Whois server	String
SRSGwWhoisIPv6	The IPv6 address or addresses to the SRS GW Whois server	String

22.4 Outcome(s)

The format of the responses MUST follow the specification outlined in section 24, "Format Specification", and all responses must be identical.

22.5 Environmental needs

- Whois client software that preserves line endings in the response, e.g. Netcat
- IPv4 connectivity
- IPv6 connectivity (if available)
- List of IP addresses (IPv4 and IPv6, respectively) from resolving <whois.nic.TLD> using the delegation data, where <WhoisIP> is one such address.
- <CharEncoding>, character encoding declaration from pdtwhois.xml file.
- Test node outside the country of the SRS GW <TestNodeA> to be used to test TLD SRS Whois server from.

- Test node inside the country of the SRS GW <TestNodeB> to be used to test SRS GW Whois server from.

22.6 Special procedural requirements

Abort the test if any Whois query takes longer than 10 seconds.

22.7 Intercase dependencies

This test has no intercase dependencies.

22.8 Ordered description of steps to be taken to execute the test case

1. Make two queries from TestNodeA to the TLD SRS Whois server using the client software, i.e. to all IP addresses found.
   whois -h **<WhoisIP>** "nameserver **<WhoisTestNameServerName>**"
   whois -h **<WhoisIP>** "nameserver **<WhoisTestNameServerIP>**"
2. Make two queries from TestNodeB to the SRS GW Whois server using the client software, i.e. to all IP addresses submitted.
   whois -h **<WhoisIP>** "nameserver **<WhoisTestNameServerName>**"
   whois -h **<WhoisIP>** "nameserver **<WhoisTestNameServerIP>**"
3. Verify that the Whois server returns responses to the two queries. A response means at least some text is returned.
4. Both queries (WhoisTestNameServerName or WhoisTestNameServerIP) for all queried Whois servers MUST give a response that matches the following steps and requirements.
   a. Verify that the character encoding of the responses match <CharEncoding>.
   b. Verify that the response is identical to all other responses to the same query.
   c. Verify that the response is a "Name server reply type 1" or "Name server reply type 2" as defined in the format specification.
   d. Verify that response conforms to the format in the specification. *This verification includes, but is not limited to, verification of ROID format and registration of ROID Suffix, mandatory fields, field names, order of fields, status codes, and data format. See section 24 for all details.*
   e. If the response is "Name server reply type 1" and the query is for WhoisTestNameServerName do the following step:
      i. Verify that the "Server Name" field field exactly matches WhoisTestNameServerName in all returned Name Server objects.
   f. If the response is "Name server reply type 1" and the query is for WhoisTestNameServerIP do the following step:
      i. If the "IP Address" field field is present, verify that at least one such field exactly matches WhoisTestNameServerIP.
      ii. Repeat the previous verification for every Name Server object in the response.
   g. If the response is "Name server reply type 2" do the following step:
      i. Extract the ROID from the first "Roid Line" as defined in the format specification.
      ii. Make a new whois query, now using the extracted ROID in the query string ("roid <ROID>").
      iii. Verify that the new response is a "Name server reply type 1" as defined in the format specification.
      iv. Verify that the new response conforms to the format in the specification.
      v. Verify the response using step e or f above.

Criteria for PASS:

- The Whois outputs are encoded in ASCII, and
- The Whois service is available on IPv4 (always) and IPv6 (if IPv6 connectivity is available in the local area test node), and
- All verifications (under steps 3 and 4a-g) are successful.

Criteria for WARN:

- The Whois outputs are not encoded in ASCII, and
- The Whois output was successfully converted to UTF-8, and
- All other criteria for PASS are successful.

Criteria for FAIL:

- The encoding of the Whois output cannot be converted to UTF-8, or
- At least one other criteria for PASS is unsuccessful.

23. General

23.1 Glossary

The glossary is available in the Master Test Plan.

23.2 Document change procedures

Document change procedures are documented in the Master Test Plan.

24. Appendix: Format Specification

This appendix defines valid Whois replies from a RST point of view. This specification has been derived from **Registry-Agreement**, **RDDS-Advisory, AWIP**, **ROID-Advisory**, **RDDS-Consistent-Labeling-Policy** and **Temp-Spec-for-gTLD-data** listed in the reference list above. All references found below (in bold face) are found in the reference list above.

24.1 Terminology

The terms "MAY", "MUST", "SHOULD" and "MUST NOT" are used to indicate the requirement level in accordance to **RFC2119**.

24.2 Limitation

This specification is limited to what is relevant for the RST Whois TCs, the format of the Whois port 43 output and the format of Web whois port 80/443 output.

Requirements on the output not tested by the RST TCs are not included in the specification below.

This appendix does not replace the Test Case section that in detail defines which types of queries are issued, what the query string is and what data, in contrast to format, expected to be found in the reply.

This specification does not consider that some Registry Operators may be permitted to provide redacted RDDS output as specified in **RDDS-Consistent-Labeling-Policy**.

24.3 Overview

At a low level the Whois output is interpreted as a sequence of characters that make up a sequence of lines. Line structure is described in detail in the "Characters and line structure" chapter.

At a high level the Whois output is interpreted as a tree where the leaves are lines and nodes are called "sections". The "Section types" chapter defines all valid "section types". The "Line types" chapter defines all valid "line types".

The entire Whois output MUST conform syntactically to a top-level section type. Additionally, there is a single semantic rule that the Whois output MUST follow, and this rule is defined together with the U-label data type in the "Data types" chapter.

24.4 Characters and line structure

The Whois port 43 output SHOULD be limited to the ASCII (US-ASCII) character set and SHOULD be encoded in ASCII (US-ASCII).

If the Whois port 43 output contains any character (code point) outside the ASCII character set, then the output SHOULD be encoded in UTF-8.

The Web whois port 80/443 output MUST follow the same conventions as Whois port 43 output except for the limitation of the character set and character encoding.
The Web whois port 80/443 output MUST follow the standards of HTTP and HTML for how to declare character set and character encoding.

When rendered using a modern web browser the Web whois port 80/443 output MUST contain a section corresponding to the Whois port 43 output that can be copied as a single unit and pasted into a text file.

The Whois output is a sequence of CRLF terminated lines (i.e. Unicode sequence "U+000D U+000A".) The line terminator is not considered to be part of the line.

Spaces (U+0020) MAY be present within a line. Any other whitespace characters (e.g. Character Tabulation (U+0009)) MUST NOT be present within a line.

Up to nine characters of leading space MAY be present at the beginning of a line. Leading space is not considered to be part of the line.

Trailing space MUST NOT be present at the end of a line unless the *line type* is an *empty field* as defined below under *line types*.

24.5 Section types

A *section* is a sequence of lines.

A *section type* is a section that matches a certain "section description".

A *section description* is an ordered list of "named field" specifications and "subsection" specifications.

A *subsection* is a section within another section. A "subsection" is specified using the format "(section name) [quantifier]" where:

- "section name" refers to a specific line type or section type
- "quantifier" is described in the "Quantifiers" chapter below

A *named field* is a "non-empty field" or an "empty field" where the "field key" value exactly matches a specific string. A "named field" is specified using the format "Key: [quantifier; data type]" where:

- "Key" is the string that the "field key" value MUST match (case sensitively)
- "quantifier" is described in the "Quantifiers" chapter below
- "data type" is reference to the format the "field value" MUST conform to

The terms "non-empty field" and "empty field" are defined in the "Line types" chapter, and all data types are defined in the "Data types" chapter – sometimes referring to other specifications.

24.5.1 Whois output

24.5.1.1 Domain Name Object queries

The Whois output in reply to Domain Name Object queries MUST in its entirety conform to the following top-level section type:

- (Domain name reply) [required]

24.5.1.2 Name Server Object queries

The Whois output in reply to Name Server Object queries MUST in its entirety conform to one of the following top-level section types:

- (Name server reply type 1) [required]
- (Name server reply type 2) [required]

24.5.1.3 Registrar Object queries

The Whois output in reply to Registrar Object queries MUST in its entirety conform to the following top-level section type:

- (Registrar reply) [required]

24.5.1.4 Other queries

In RST, no other query types will be issued.

24.5.2 Registrar reply

1. (Registrar details section) [required]
2. (Subsequent registrar details section) [optional-repeatable]
3. (Last update footer) [required]
4. (AWIP footer) [optional-free]
5. (Legal disclaimer) [required]

24.5.3 Domain name reply

1. (Domain name details section) [required]
2. (Subsequent domain name details section) [optional-repeatable]
3. (Last update footer) [required]
4. (AWIP footer) [required]
5. (Legal disclaimer) [required]

24.5.4 Name server reply type 1

1. (Name server details section) [required]
2. (Subsequent name server details section) [optional-repeatable]
3. (Last update footer) [required]
4. (AWIP footer) [optional-free]
5. (Legal disclaimer) [required]

24.5.5 Name server reply type 2

1. (Multiple name servers section) [required]
2. (Last update footer) [required]
3. (AWIP footer) [optional-free]
4. (Legal disclaimer) [required]

24.5.6 Subsequent registrar details section

1. (Empty line) [required]
2. (Registrar details section) [required]

24.5.7 Registrar details section

1. Registrar: [required; type "postal line"]
2. Street: [repeatable; type "postal line"]

3. City: [required; type "postal line"]
4. State/Province: [optional-constrained; type "postal line"]
5. Postal Code: [optional-constrained; type "postal code"]
6. Country: [required; type "country code"]
7. (Phone number section) [repeatable]
8. (Fax number section) [required]
9. Email: [repeatable; type "email address"]
10. Registrar WHOIS Server: [optional-constrained; type "hostname"]
11. Registrar URL: [required; type "http url"]
12. (Admin contact section) [optional-repeatable]
13. (Technical contact section) [optional-repeatable]
14. (Registrar object additional fields section) [optional-free]

24.5.8 Admin contact section

1. Admin Contact: [required; type "postal line"]
2. (Phone number section) [repeatable]
3. (Fax number section) [required]
4. Email: [repeatable; type "email address"]

24.5.9 Technical contact section

1. Technical Contact: [required; type "postal line"]
2. (Phone number section) [repeatable]
3. (Fax number section) [required]
4. Email: [repeatable; type "email address"]

24.5.10 Phone number section

1. Phone Number: [required; type "phone number"]
2. Phone Ext: [optional-free; type "token"]

24.5.11 Fax number section

Either one of:

- (Fax number section type A) [repeatable]
- (Fax number section type B) [required]
- (Fax number section type C) [required]

24.5.12 Fax number section type A

1. Fax Number: [required; type "phone number"]
2. Fax Ext: [optional-free; type "token"]

24.5.13 Fax number section type B

1. Fax Number: [empty-constrained; type "void"]
2. Fax Ext: [optional-free; type "token"]

24.5.14 Fax number section type C

1. Fax Number: [omitted-constrained; type "void"]

24.5.15 Subsequent domain name details section

1. (Empty line) [required]
2. (Domain name details section) [required]

24.5.16 Domain name details section

1. Domain Name: [required; type "hostname"]
2. Internationalized Domain Name: [optional-free; type "u-label"]
3. Registry Domain ID: [required; type "ROID"]
4. Registrar WHOIS Server: [optional-constrained; type "hostname"]
5. Registrar URL: [required; type "http url"]
6. Updated Date: [optional-constrained; type "time stamp"]
7. Creation Date: [required; type "time stamp"]
8. Registry Expiry Date: [required; type "time stamp"]
9. Registrar Registration Expiration Date: [optional-constrained; type "time stamp"]
10. Registrar: [required; type "token"]
11. Registrar IANA ID: [required; type "positive integer"]
12. (Domain name details subsection 1): [required]
13. URL of the ICANN Whois Inaccuracy Complaint Form: [required; type "inaccuracy complaint url"]

24.5.17 Domain name details subsection 1

Either one of:

- (Domain name details subsection 2): [required]
- (Domain name details subsection 3): [required]

24.5.18 Domain name details subsection 2

1. (Domain name details subsection 4): [required]
2. (Domain abuse contact section): [required]

24.5.19 Domain name details subsection 3

1. (Domain abuse contact section): [required]
2. (Domain name details subsection 4): [required]

24.5.20 Domain name details subsection 4

1. Reseller: [optional-constrained; type "token"]
2. Domain Status: [repeatable; type "domain status"]
3. Registry Registrant ID: [optional-free; type "ROID-or-redacted"]
4. Registrant Name: [optional-free; type "postal line"]
5. Registrant Organization: [optional-constrained; type "postal line"]
6. Registrant Street: [optional-repeatable; type "postal line"]
7. Registrant City: [optional-free; type "postal line"]
8. Registrant State/Province: [optional-constrained; type "postal line"]
9. Registrant Postal Code: [optional-free; type "postal-code-or-redacted"]
10. Registrant Country: [required; type "country code"]
11. Registrant Phone: [optional-free; type "phone-number-or-redacted"]
12. Registrant Phone Ext: [optional-free; type "token-or-redacted"]
13. Registrant Fax: [optional-free; type "phone-number-or-redacted"]
14. Registrant Fax Ext: [optional-free; type "token-or-redacted"]

15. Registrant Email: [required; type "email-web-or-redacted"]
16. Registry Admin ID: [optional-free; type "ROID-or-redacted"]
17. Admin Name: [optional-free; type "postal line"]
18. Admin Organization: [optional-free; type "postal line"]
19. Admin Street: [optional-repeatable; type "postal line"]
20. Admin City: [optional-free; type "postal line"]
21. Admin State/Province: [optional-free; type "postal line"]
22. Admin Postal Code: [optional-free; type "postal-code-or-redacted"]
23. Admin Country: [optional-free; type "country-code-or-redacted"]
24. Admin Phone: [optional-free; type "phone-number-or-redacted"]
25. Admin Phone Ext: [optional-free; type "token-or-redacted"]
26. Admin Fax: [optional-free; type "phone-number-or-redacted"]
27. Admin Fax Ext: [optional-free; type "token-or-redacted"]
28. Admin Email: [required; type "email-web-or-redacted"]
29. Registry Tech ID: [optional-free; type "ROID-or-redacted"]
30. Tech Name: [optional-free; type "postal line"]
31. Tech Organization: [optional-free; type "postal line"]
32. Tech Street: [optional-repeatable; type "postal line"]
33. Tech City: [optional-free; type "postal line"]
34. Tech State/Province: [optional-free; type "postal line"]
35. Tech Postal Code: [optional-free; type "postal-code-or-redacted"]
36. Tech Country: [optional-free; type "country-code-or-redacted"]
37. Tech Phone: [optional-free; type "phone-number-or-redacted"]
38. Tech Phone Ext: [optional-free; type "token-or-redacted"]
39. Tech Fax: [optional-free; type "phone-number-or-redacted"]
40. Tech Fax Ext: [optional-free; type "token-or-redacted"]
41. Tech Email: [required; type "email-web-or-redacted"]
42. (Billing contact section) [optional-free]
43. (Name server section) [required]
44. DNSSEC: [required; type "dnssec"]
45. (Domain name object additional fields section) [optional-free]

24.5.21 Domain abuse contact section

1. Registrar Abuse Contact Email: [required; type "email address"]
2. Registrar Abuse Contact Phone: [required; type "phone number"]

24.5.22 Billing contact section

1. Registry Billing ID: [optional-free; type "ROID-or-redacted"]
2. Billing Name: [optional-free; type "postal line"]
3. Billing Organization: [optional-free; type "postal line"]
4. Billing Street: [optional-repeatable; type "postal line"]
5. Billing City: [optional-free; type "postal line"]
6. Billing State/Province: [optional-free; type "postal line"]
7. Billing Postal Code: [optional-free; type "postal-code-or-redacted"]
8. Billing Country: [optional-free; type "country-code-or-redacted"]
9. Billing Phone: [optional-free; type "phone-number-or-redacted"]
10. Billing Phone Ext: [optional-free; type "token-or-redacted"]
11. Billing Fax: [optional-free; type "phone-number-or-redacted"]
12. Billing Fax Ext: [optional-free; type "token-or-redacted"]

13. Billing Email: [required; type "email-web-or-redacted"]

24.5.23 Name server section

Either one of:

- (Name server section type A) [repeatable]
- (Name server section type B) [required]
- (Name server section type C) [required]

24.5.24 Name server section type A

1. Name Server: [required; type "hostname"]
2. (IP Address section) [repeatable]

24.5.25 Name server section type B

1. Name Server: [empty-constrained; type "void"]

24.5.26 Name server section type C

1. Name Server: [omitted-constrained; type "void"]

24.5.27 IP Address section

1. IP Address: [optional-not-empty; type "ip address"]

24.5.28 Multiple name servers section

1. (Multiple name servers line) [required]
2. (ROID line) [required]
3. (ROID line) [repeatable]

24.5.29 Subsequent name server details section

1. (Empty line) [required]
2. (Name server details section) [required]

24.5.30 Name server details section

1. Server Name: [required; type "hostname"]
2. IP Address: [optional-repeatable; type "ip address"]
3. Registrar: [optional-constrained; type "postal line"]
4. Registrar WHOIS Server: [optional-constrained; type "hostname"]
5. Registrar URL: [optional-constrained; type "http url"]
6. (Name server object additional fields section) [optional-free]

24.5.31 Registrar object additional fields section

1. (Registrar object additional field) [repeatable]

24.5.32 Registrar object additional field

Either one of:

- (Non-empty field) [required]
- (Empty field) [required]

The "field key" in such field MUST NOT match any of the field key strings listed in "Forbidden Registrar object additional field names".

24.5.33 Domain name object additional fields section

1. (Domain name object additional field) [repeatable]

24.5.34 Domain name object additional field

Either one of:

- (Non-empty field) [required]
- (Empty field) [required]

The "field key" in such field MUST NOT match any of the field key strings listed in "Forbidden Domain name object additional field names".

24.5.35 Name server object additional fields section

1. (Name server object additional field) [repeatable]

24.5.36 Name server object additional field

Either one of:

- (Non-empty field) [required]
- (Empty field) [required]

The "field key" in such field MUST NOT match any of the field key strings listed in "Forbidden Name server object additional field names".

24.5.37 Last update footer

1. (Empty line) [optional-repeatable max 3 times]
2. (Last update line) [required]

24.5.38 AWIP footer

1. (Empty line) [repeatable max 3 times]
2. (AWIP line) [required]

24.5.39 Legal disclaimer

1. (Empty line) [repeatable max 3 times]
2. (Non-empty line)
3. (Empty line or Non-empty line) [repeatable]

24.6 Line types

24.6.1 Empty line

An *empty line* is a line that consists of the empty string. Note: Empty lines with leading space are still considered empty.

24.6.2 Non-empty line

A *non-empty line* is a line that consists of one or more non-space characters.

24.6.3 Non-empty field

A *non-empty field* is a line that consists of the following parts:

1. A string matching type "field key"
2. The empty string or a string of type "translation clause"
3. The exact string ": " (U+003A U+0020)
4. A non-empty string – referred to as the *field value*

Example:

Domain Name: EXAMPLE.TLD

See the "Data types" chapter for definitions of "field key" and "translation clause".

24.6.4 Empty field

An *empty field* is a line that consists of the following parts:

1. A string matching type "field key"
2. The empty string or a string matching type "translation clause"
3. The exact string ":" (U+003A)
4. Optionally the exact string " " (U+0020)

Example:

Name Server:

See the "Data types" chapter for definitions of "field key" and "translation clause". A line with an *empty field* is the only type of line that MAY have a trailing space.

24.6.5 Multiple name servers line

A *multiple name servers line* consists of the following parts:

1. The exact string "Query matched more than one name server:"

24.6.6 ROID line

A *ROID line* is a line that consists of the following parts:

1. A string matching type "ROID"
2. The exact string " (" (U+0020 U+0028)
3. A string matching type "hostname"
4. The exact string ")" (U+0029)

Example:

roid1abc-example (ns1.foo.example)

24.6.7 Last update line

A *last update line* is a line that consists of the following parts:

1. The exact string ">>> Last update of "
2. The exact string "Whois" or the exact string "WHOIS"
3. The exact string " database: "

4. A string matching type "time stamp"
5. The exact string " <<<"

Two examples:

>>> Last update of Whois database: 2014-11-14T12:58:01Z <<<
>>> Last update of WHOIS database: 2014-11-14T12:58:01Z <<<

See the "Data types" chapter for definitions of "time stamp".

24.6.8 AWIP line

An *AWIP line* is a line that consists of one of the following parts:

1. The exact string (quote marks excluded and no line breaks) "For more information on Whois status codes, please visit https://icann.org/epp"
2. The exact string (quote marks excluded and no line breaks) "For more information on Whois status codes, please visit https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en"

Also see **AWIP**.

24.7 Quantifiers

24.7.1 Required

If a "subsection" is declared *required*, a sequence of lines conforming to the given section description MUST appear at that position.

If a "named field" is declared as *required*, one "non-empty field" with the given "field key" value and a "field value" of the specified data type MUST appear at that position.

24.7.2 Optional-constrained

If a "named field" is declared *optional-constrained*, one of the following formats MUST be used:

1) One "empty field" with the given "field key" value MUST appear at that position (EMPTY).
2) A "field" with the given "field key" value MUST NOT appear at that position (OMITTED).
3) One "non-empty field" with the given "field key" value and a "field value" of the specified data type MUST appear at that position.

"Named fields" declared *optional-constrained* are subject to the restriction defined in "Special restrictions on 'constrained'".

24.7.3 Empty-constrained

If a "named field" is declared *empty-constrained*, the following format MUST be used:

1) One "empty field" with the given "field key" value MUST appear at that position (EMPTY).

"Named fields" declared *empty-constrained* are subject to the restriction defined in "Special restrictions on 'constrained'".

24.7.4 Omitted-constrained

If a "named field" is declared *omitted-constrained*, the following format MUST be used:

1) A "field" with the given "field key" value MUST NOT appear at that position (OMITTED).

"Named fields" declared *omitted-constrained* are subject to the restriction defined in "Special restrictions on 'constrained'".

24.7.5 Special restrictions on "constrained"

It is not permitted to mix "empty fields" (EMPTY) with omitted (OMITTED) fields for "named fields" declared as optional-constrained, empty-constrained or omitted-constrained in the same Whois output.

24.7.6 Optional-free

If a "subsection" is declared *optional-free*, a sequence of lines conforming to the description of the given section MAY appear once at that position.

If a "named field" is declared *optional-free*, one of the following formats MUST be used:

1) One "empty field" with the given "field key" value MUST appear at that position.
2) A "field" with the given "field key" value MUST NOT appear at that position.
3) One "non-empty field" with the given "field key" value and a "field value" of the specified data type MUST appear at that position.

Optional-free fields may, in contrast to *optional-constrained* fields, mix formats 1 and 2 in the same Whois output.

24.7.7 Optional-not-empty

If a "named field" is declared *optional-not-empty*, one of the following formats MUST be used:

1) A "field" with the given "field key" value MUST NOT appear at that position.
2) One "non-empty field" with the given "field key" value and a "field value" of the specified data type MUST appear at that position.

24.7.8 Repeatable

If a "subsection" is declared *repeatable*, a sequence of lines conforming to the given section description MUST appear one or more times the given position. The number of occurrences MUST conform to the upper bound if one is specified.

If a "named field" is declared as *repeatable*, one or more instances of "non-empty field" with the given "field key" value and a "field value" of the specified data type MUST appear at that position. The number of occurrences MUST conform to the upper bound if one is specified.

24.7.9 Optional-repeatable

If a "subsection" is declared *optional-repeatable*, a sequence of lines conforming to the given section description quantified as either *optional-free* or *repeatable* MUST appear at that position.

If a "named field" is declared as *optional-repeatable*, one of the following formats MUST be used:

1. The "named field" is treated as declared *optional-free*.
2. The "named field" is treated as declared *repeatable*.

24.8 Data types

24.8.1 Translation clause

A *translation clause* consists of the following parts:

1. the exact string " (" (U+0020 U+0028)
2. a key translation [type "key translation"]
3. one or more translations separated by "/" (U+002F)
4. the exact string ")" (U+0029)

24.8.2 Key translation

A *key translation* is a string that MUST NOT begin or end with a space (U+0020) and MUST NOT contain any parentheses (U+0028, U+0029).

24.8.3 Field key

A *field key* is a case sensitive string that MUST NOT contain ":" (U+003A).

24.8.4 Redact String

A *Redact String* is a string that MUST meet one of the following specifications:

1. The exact string (quote marks excluded) “REDACTED FOR PRIVACY” in upper, lower or mixed case.
2. A string that is substantially similar to the string in 1.

The *Redact String* data types meets the requirements in **Temp-Spec-for-gTLD-data**, appendix A, section 2.2.

24.8.5 ROID

A *ROID* MUST follow the definition of "roidType" in **RFC5730**, section 4.2, p. 58.

The requirement of the *ROID Suffix* MUST be met.

24.8.6 ROID Suffix

A *ROID* is split in two parts by the single, mandatory HYPHEN (U+002D). The string after the HYPHEN is here defined as *ROID Suffix*.

The *ROID Suffix* MUST be a registered *EPP Repository Identifier* found in **IANA-Repository** (see **ROID-Advisory**).

24.8.7 ROID-or-redacted

A *ROID-or-redacted* MUST either meet the definition of *Redact String* or *ROID*.

24.8.8 Hostname

A *hostname* (see **RFC952** and **RFC1123**) MUST match the following Perl compatible Regular Expression: '^([a-zA-Z0-9]([a-zA-Z0-9-]*[a-zA-Z0-9])*\.){1,}[a-zA-Z]([a-zA-Z0-9-]*[a-zA-Z0-9])\.?$'

A *label* within a *hostname* MUST NOT exceed 63 characters, where the *labels* are defined as the strings resulting from splitting the *hostname* by the dots.

The total length of a *hostname* MUST NOT exceed 254 characters excluding any final dot.

Example:
ns1.xn--caf-dma.example
ns1.example.example.

The *hostname* in an email address MUST NOT have the optional trailing dot.

24.8.9 Time stamp

A *Time stamp* MUST be defined as "date-time" in **RFC3339** (section 5.6, p. 8) and "time-offset" MUST be "Z".

24.8.10 U-label

A *U-label* MUST follow the definition in **RFC5890**, section 2.3.2.1, p. 12.

The "Internationalized Domain Name" field value and the "Domain Name" field value in a "Domain name details section" MUST be equivalent according to **RFC5890**.

24.8.11 HTTP URL

An *http url* is a URL for HTTP or HTTPS and MUST be given as defined in **RFC7230**, sections 2.7.1 and 2.7.2, respectively.

24.8.12 Token

A *Token* MUST be as defined in **XML-Schema**, section 3.3.2.

24.8.13 Token-or-redacted

A *Token-or-redacted* MUST either meet the definition of *Redact String* or *Token*.

24.8.14 Positive integer

A *Positive integer* is a string matching the following Perl compatible Regular Expression:
'^[1-9][0-9]*$'

A *positive integer* is a decimal number.

24.8.15 Domain status

A *domain status* consists of the following parts:
1) a domain status code [type "domain status code"]
2) at least one and no more than nine spaces (U+0020)
3) the exact string "https://icann.org/epp#"
4) the same domain status code as in step 1 above

Example:
inactive https://icann.org/epp#inactive

There is one exception to step 4 above. If the domain status code in step 1 is "ok" then the code in the URL fragment can be in upper case instead, i.e. "OK". The following two formats are legal according to this exception:

ok https://icann.org/epp#ok
ok https://icann.org/epp#OK

24.8.16 Domain status code

A *domain status code* MUST be one of the EPP status codes listed in type "statusValueType" in **RFC5731** (section 4, p. 38) or as "Status Values" in **RFC3915** (section 3.1, p. 6), also listed here:

- addPeriod
- autoRenewPeriod
- clientDeleteProhibited
- clientHold
- clientRenewProhibited
- clientTransferProhibited
- clientUpdateProhibited
- inactive
- ok
- pendingCreate
- pendingDelete
- pendingRenew
- pendingRestore
- pendingTransfer
- pendingUpdate
- redemptionPeriod
- renewPeriod
- serverDeleteProhibited
- serverHold
- serverRenewProhibited
- serverTransferProhibited
- serverUpdateProhibited
- transferPeriod

24.8.17 Postal line

A *Postal line* MUST follow the definition for "postalLineType" in **RFC5733**, section 4, p. 30.

24.8.18 Postal code

A *Postal code* MUST follow the definition for "pcType" in **RFC5733**, section 4, p. 30.

24.8.19 Postal-code-or-redacted

A *Postal-code-or-redacted* MUST meet the definition of *Redact String* or *Postal code*.

24.8.20 Country code

A *country code* is a two-letter string that MUST match the following Perl compatible Regular Expression:

'^[a-zA-Z]{2}$'

Informational: The codes in ISO 3166-1, alpha-2 list, are used as country codes. See **ISO3166-1-ISO** and **ISO3166-1-Wikipedia** for details.

24.8.21 Country-code-or-redacted

A *Country-code-or-redacted* MUST meet the definition of *Redact String* or *Country code*.

24.8.22 Phone number

See "e164StringType" in **RFC5733**, section 4, p. 30.

24.8.23 Phone-number-or-redacted

A *Phone-number-or-redacted* MUST meet the definition of *Redact String* or *Phone number*.

24.8.24 Email address

An "email address" as the strict definition of email address for Internet mail by **RFC5322**:

email_address	=	localpart "@" domain
localpart	=	atext *("." atext)
atext	=	as defined in **RFC5322**, sec 3.2.3, p. 13
domain	=	hostname as defined in this document

The hostname part MUST be without the trailing dot.

24.8.25 Email redact string

An *Email Redact String* is a string that MUST meet one of the following specifications:

1. The string (quote marks excluded and no line breaks) "Please query the RDDS service of the Registrar of Record identified in this output for information on how to contact the Registrant, Admin, or Tech contact of the queried domain name." in upper, lower or mixed case.
2. A string that is substantially similar to the string in 1.

The *Email Redact String* data types meets the requirements in **Temp-Spec-for-gTLD-data**, appendix A, section 2.5.2.

24.8.26 Email-web-or-redacted

An *Email-web-or-redacted* MUST meet the definition of *Email Address*, *HTTP URL* or *Email Redact String*.

24.8.27 DNSSEC

One of the following exact strings:

- signedDelegation
- unsigned

24.8.28 IP address

An *ip address* is either one *IPv4 address* or one *IPv6 address*.

24.8.29 IPv4 address

IPv4 addresses are represented as four decimal octets separated with single dots, in dot-decimal notation. The IPv4 address representation contains decimal digits and dots only. See also **RFC791** and **Dot-Decimal**.

The IPv4 address MUST match the following definition:

IPv4 address =	octet "." octet "." octet "." octet	
octet	=	octet-zero / octet-positive
octet-zero	=	"0"
octet-positive =	*Positive integer* < *256*	

Positive integer as defined in this document.

Example:

- 192.0.2.3

24.8.30 IPv6 address

IPv6 addresses MUST be represented as described in **RFC4291**, section 2.2.

Example:

- 2001:db8::5

24.8.31 Void

This is a placeholder for data type when a *named field* is defined to be an "empty field" or an "empty line".

24.8.32 Forbidden Registrar object additional field names

The following field names are forbidden in Registrar object additional fields:

- Registrar Name
- Street
- City
- State/Province
- Postal Code
- Country
- Phone Number
- Email
- WHOIS Server
- Referral URL
- Admin Contact
- Technical Contact
- Fax Number
- Registrar WHOIS Server
- Registrar URL
- Registrar

24.8.33 Forbidden Domain name object additional field names

The following field names are forbidden in Domain name object additional fields:

- Domain Name
- Domain ID
- WHOIS Server
- Referral URL
- Updated Date
- Creation Date
- Registry Expiry Date
- Sponsoring Registrar
- Sponsoring Registrar IANA ID
- Domain Status
- Registrant ID
- Registrant Name
- Registrant Organization
- Registrant Street
- Registrant City
- Registrant State/Province
- Registrant Postal Code
- Registrant Country
- Registrant Phone
- Registrant Phone Ext
- Registrant Fax
- Registrant Fax Ext
- Registrant Email
- Admin ID
- Admin Name
- Admin Organization
- Admin Street
- Admin City
- Admin State/Province
- Admin Postal Code
- Admin Country
- Admin Phone
- Admin Phone Ext
- Admin Fax
- Admin Fax Ext
- Admin Email
- Tech ID
- Tech Name
- Tech Organization
- Tech Street
- Tech City

- Tech State/Province
- Tech Postal Code
- Tech Country
- Tech Phone
- Tech Phone Ext
- Tech Fax
- Tech Fax Ext
- Tech Email
- DNSSEC
- Name Server
- IP Address
- Registry Domain ID
- Registrar WHOIS Server
- Registrar URL
- Registrar
- Registrar IANA ID
- Registry Registrant ID
- Registry Admin ID
- Registry Tech ID
- URL of the ICANN Whois Inaccuracy Complaint Form

24.8.34 Forbidden Name server object additional field names

The following field names are forbidden in Name server object additional fields:

- Server Name
- IP Address
- Registrar
- WHOIS Server
- Referral URL
- Registrar WHOIS Server
- Registrar URL

24.8.35 Inaccuracy complaint URL

An *inaccuracy complaint url* MUST be the exact string "https://www.icann.org/wicf/".