-----Bonjour-----320.10.80-----

Apache License

Version 2.0, January 2004

http:11www,apache.org/licenses/ (http:11www,apache.org/licenses/)

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBU'TION

1. Definitions,

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License,

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counter claim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and

b. You must cause any modified files to carry prominent notices stating that You changed the files; and

c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions, Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor

regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License,

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

-----gSOAP 2.7.6C-----

gSOAP Public License

Version 1.3a

The gSOAP public license is derived from the Mozilla Public License (MPL1.1). The sections that were deleted from the original MPL1.1 text are 1.0.1, 2.1.(c),(d), 2.2.(c),(d), 8.2.(b), 10, and 11. Section 3.8 was added. The modified sections are 2.1.(b), 2.2.(b), 3.2 (simplified), 3.5 (deleted the last sentence), and 3.6 (simplified).

1 DEFINITIONS.

1.0.1.

1.1. “Contributor” means each entity that creates or contributes to the creation of Modifications.

1.2. “Contributor Version” means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. “Covered Code” means the Original Code, or Modifications or the combination of the Original Code, and Modifications, in each case including portions thereof.

1.4. “Electronic Distribution Mechanism” means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. “Executable” means Covered Code in any form other than Source Code.

1.6. “Initial Developer” means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. “Larger Work” means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. “License” means this document.

1.8.1. ”Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. “Modifications” means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code, or previous Modifications.

1.10. “Original Code” means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. “Patent Claims” means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. “Source Code” means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor’s choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. “You” (or “Your”) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, “You” includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2 SOURCE CODE LICENSE.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell (“offer to sell and import”) the Original Code, Modifications, or portions thereof, but solely to the extent that any such patent is reasonably necessary to enable You to utilize, alone or in combination with other software, the Original Code, Modifications, or any combination or portions thereof.

(c)

(d)

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royaltyfree, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to make, have made, use and sell ("offer to sell and import") the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to utilize, alone or in combination with other software, the Contributor Version (or portions thereof).

(c)

(d)

3 DISTRIBUTION OBLIGATIONS.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification created by You will be provided to the Initial Developer in Source Code form and are subject to the terms of the License.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters.

(a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs. If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations. Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. If you distribute executable versions containing Covered Code as part of a product, you must reproduce the notice in Exhibit B in the documentation and/or other materials provided with the product.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the LargerWork as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

3.8. Restrictions.

You may not remove any product identification, copyright, proprietary notices or labels from gSOAP.

4 INABILITY TO COMPLY DUE TO STATUTE OR REGULATION.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5 APPLICATION OF THIS LICENSE.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6 VERSIONS OF THE LICENSE.

6.1. New Versions.

Grantor may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License.

6.3. DerivativeWorks.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrase " gSOAP" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the gSOAP Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7 DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIEDWARRANTIES OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT

OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND ANY WARRANTY THAT MAY ARISE BY REASON OF TRADE USAGE, CUSTOM, OR COURSE OF DEALING. WITHOUT LIMITING THE FOREGOING, YOU ACKNOWLEDGE THAT THE SOFTWARE IS PROVIDED "AS IS" AND THAT THE AUTHORS DO NOTWARRANT THE SOFTWAREWILL RUN UNINTERRUPTED OR ERROR FREE. LIMITED LIABILITY THE ENTIRE RISK AS TO RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. UNDER NO CIRCUMSTANCES WILL THE AUTHORS BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE WHATSOEVER, WHETHER BASED ON CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, ARISING OUT OF OR IN ANY WAY RELATED TO THE SOFTWARE, EVEN IF THE AUTHORS HAVE BEEN ADVISED ON THE POSSIBILITY OF SUCH DAMAGE OR IF SUCH DAMAGE COULD HAVE BEEN REASONABLY FORESEEN, AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY EXCLUSIVE REMEDY PROVIDED. SUCH LIMITATION ON DAMAGES INCLUDES, BUT IS NOT LIMITED TO, DAMAGES FOR LOSS OF GOODWILL, LOST PROFITS, LOSS OF DATA OR SOFTWARE, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION OR IMPAIRMENT OF OTHER GOODS. IN NO EVENTWILL THE AUTHORS BE LIABLE FOR THE COSTS OF PROCUREMENT

OF SUBSTITUTE SOFTWARE OR SERVICES. YOU ACKNOWLEDGE THAT THIS SOFTWARE IS NOT DESIGNED FOR USE IN ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS SUCH AS OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR CONTROL, OR LIFE-CRITICAL APPLICATIONS. THE AUTHORS EXPRESSLY DISCLAIM ANY LIABILITY RESULTING FROM USE OF THE SOFTWARE IN ANY SUCH ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS AND ACCEPTS NO LIABILITY IN RESPECT OF ANY ACTIONS OR CLAIMS BASED ON THE USE OF THE SOFTWARE IN ANY SUCH ONLINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS BY YOU. FOR PURPOSES OF THIS PARAGRAPH, THE TERM “LIFE-CRITICAL APPLICATION” MEANS AN APPLICATION IN WHICH THE FUNCTIONING OR MALFUNCTIONING OF THE SOFTWARE MAY RESULT DIRECTLY OR INDIRECTLY IN PHYSICAL INJURY OR LOSS OF HUMAN LIFE. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8 TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant’s Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9 LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10 U.S. GOVERNMENT END USERS.

11 MISCELLANEOUS.

12 RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

EXHIBIT A.

"The contents of this file are subject to the gSOAP Public License Version 1.3 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://genivia.com/Products/gsoap/license.pdf

More information on licensing options, support contracts, and consulting can be found at

http://genivia.com/Products/gsoap/contract.html

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code of the gSOAP Software is: stdsoap.h, stdsoap2.h, stdsoap.c, stdsoap2.c, stdsoap.cpp, stdsoap2.cpp, soapcpp2.h, soapcpp2.c, soapcpp2 lex.l, soapcpp2 yacc.y, error2.h, error2.c, symbol2.c, init2.c, soapdoc2.html, and soapdoc2.pdf, httpget.h, httpget.c, stl.h, stldeque.h, stllist.h, stlvector.h, stlset.h.

The Initial Developer of the Original Code is Robert A. van Engelen. Portions created by Robert A. van Engelen are Copyright (C) 2001–2004 Robert A. van Engelen, Genivia inc. All Rights Reserved.

Contributor(s):

" ."

[Note: The text of this Exhibit A may differ slightly form the text of the notices in the Source Code files of the Original code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

EXHIBIT B.

"Part of the software embedded in this product is gSOAP software.

Portions created by gSOAP are Copyright (C) 2001–2004 Robert A. van Engelen, Genivia, Inc. All Rights Reserved.

THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANYWAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

-----OpenSSL 1.0.1i-----

LICENSE ISSUES

==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts.

OpenSSL License

```
/* ====================================================================
 * Copyright (c) 1998-2016 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
```

*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.   IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).   This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to.   The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code.   The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).

*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.   IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.   i.e. this code cannot simply be

* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

-----wpa_supplicant 2.0-----

WPA Supplicant
==============

Copyright (c) 2003-2016, Jouni Malinen <j@w1.fi> and contributors
All Rights Reserved.

This program is licensed under the BSD license (the one with advertisement clause removed).

If you are submitting changes to the project, please see CONTRIBUTIONS file for more instructions.

License

This software may be distributed, used, and modified under the terms of BSD license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name(s) of the above-listed copyright holder(s) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Features

Supported WPA/IEEE 802.11i features:

- WPA-PSK ("WPA-Personal")
- WPA with EAP (e.g., with RADIUS authentication server) ("WPA-Enterprise")
  Following authentication methods are supported with an integrate IEEE 802.1X
  Supplicant:
  * EAP-TLS
  * EAP-PEAP/MSCHAPv2 (both PEAPv0 and PEAPv1)
  * EAP-PEAP/TLS (both PEAPv0 and PEAPv1)
  * EAP-PEAP/GTC (both PEAPv0 and PEAPv1)
  * EAP-PEAP/OTP (both PEAPv0 and PEAPv1)
  * EAP-PEAP/MD5-Challenge (both PEAPv0 and PEAPv1)
  * EAP-TTLS/EAP-MD5-Challenge
  * EAP-TTLS/EAP-GTC
  * EAP-TTLS/EAP-OTP
  * EAP-TTLS/EAP-MSCHAPv2
  * EAP-TTLS/EAP-TLS
  * EAP-TTLS/MSCHAPv2
  * EAP-TTLS/MSCHAP
  * EAP-TTLS/PAP
  * EAP-TTLS/CHAP
  * EAP-SIM
  * EAP-AKA

* EAP-AKA'
* EAP-PSK
* EAP-PAX
* EAP-SAKE
* EAP-IKEv2
* EAP-GPSK
* EAP-pwd
* LEAP (note: requires special support from the driver for IEEE 802.11 authentication)

(following methods are supported, but since they do not generate keying material, they cannot be used with WPA or IEEE 802.1X WEP keying)

* EAP-MD5-Challenge
* EAP-MSCHAPv2
* EAP-GTC
* EAP-OTP

- key management for CCMP, TKIP, WEP104, WEP40
- RSN/WPA2 (IEEE 802.11i)
  * pre-authentication
  * PMKSA caching

Supported TLS/crypto libraries:
- OpenSSL (default)
- GnuTLS

Internal TLS/crypto implementation (optional):
- can be used in place of an external TLS/crypto library
- TLSv1
- X.509 certificate processing
- PKCS #1
- ASN.1
- RSA
- bignum
- minimal size (ca. 50 kB binary, parts of which are already needed for WPA; TLSv1/X.509/ASN.1/RSA/bignum parts are about 25 kB on x86)

Requirements

Current hardware/software requirements:

- Linux kernel 2.4.x or 2.6.x with Linux Wireless Extensions v15 or newer
- FreeBSD 6-CURRENT
- NetBSD-current
- Microsoft Windows with WinPcap (at least WinXP, may work with other versions)
- drivers:

  Linux drivers that support cfg80211/nl80211. Even though there are number of driver specific interface included in wpa_supplicant, please note that Linux drivers are moving to use generic wireless configuration interface driver_nl80211 (-Dnl80211 on wpa_supplicant command line) should be the default option to start with before falling back to driver specific interface.

  Linux drivers that support WPA/WPA2 configuration with the generic Linux wireless extensions (WE-18 or newer). Obsoleted by nl80211.

  In theory, any driver that supports Linux wireless extensions can be used with IEEE 802.1X (i.e., not WPA) when using ap_scan=0 option in configuration file.

  Wired Ethernet drivers (with ap_scan=0)

  BSD net80211 layer (e.g., Atheros driver)
  At the moment, this is for FreeBSD 6-CURRENT branch and NetBSD-current.

  Windows NDIS
  The current Windows port requires WinPcap (http://winpcap.polito.it/).
  See README-Windows.txt for more information.

wpa_supplicant was designed to be portable for different drivers and operating systems. Hopefully, support for more wlan cards and OSes will be added in the future. See developer's documentation (http://hostap.epitest.fi/wpa_supplicant/devel/) for more information about the design of wpa_supplicant and porting to other drivers. One main goal is to add full WPA/WPA2 support to Linux wireless extensions to allow new drivers to be supported without having to implement new driver-specific interface code in wpa_supplicant.

Optional libraries for layer2 packet processing:
- libpcap (tested with 0.7.2, most relatively recent versions assumed to work,
  this is likely to be available with most distributions,

http://tcpdump.org/)
- libdnet (tested with v1.4, most versions assumed to work,
  http://libdnet.sourceforge.net/)

These libraries are _not_ used in the default Linux build. Instead, internal Linux specific implementation is used. libpcap/libdnet are more portable and they can be used by adding CONFIG_L2_PACKET=pcap into .config. They may also be selected automatically for other operating systems. In case of Windows builds, WinPcap is used by default (CONFIG_L2_PACKET=winpcap).

Optional libraries for EAP-TLS, EAP-PEAP, and EAP-TTLS:
- OpenSSL (tested with 1.0.1 and 1.0.2 versions; assumed to work with most relatively recent versions; this is likely to be available with most distributions, http://www.openssl.org/)
- GnuTLS
- internal TLSv1 implementation

One of these libraries is needed when EAP-TLS, EAP-PEAP, EAP-TTLS, or EAP-FAST support is enabled. WPA-PSK mode does not require this or EAPOL/EAP implementation. A configuration file, .config, for compilation is needed to enable IEEE 802.1X/EAPOL and EAP methods. Note that EAP-MD5, EAP-GTC, EAP-OTP, and EAP-MSCHAPV2 cannot be used alone with WPA, so they should only be enabled if testing the EAPOL/EAP state machines. However, there can be used as inner authentication algorithms with EAP-PEAP and EAP-TTLS.

See Building and installing section below for more detailed information about the wpa_supplicant build time configuration.

WPA

The original security mechanism of IEEE 802.11 standard was not designed to be strong and has proven to be insufficient for most networks that require some kind of security. Task group I (Security) of IEEE 802.11 working group (http://www.ieee802.org/11/) has worked

to address the flaws of the base standard and has in practice
completed its work in May 2004. The IEEE 802.11i amendment to the IEEE 802.11 standard was approved in June 2004 and published in July 2004.

Wi-Fi Alliance (http://www.wi-fi.org/) used a draft version of the IEEE 802.11i work (draft 3.0) to define a subset of the security enhancements that can be implemented with existing wlan hardware. This is called Wi-Fi Protected Access<TM> (WPA). This has now become a mandatory component of interoperability testing and certification done by Wi-Fi Alliance. Wi-Fi provides information about WPA at its web site (http://www.wi-fi.org/OpenSection/protected_access.asp).

IEEE 802.11 standard defined wired equivalent privacy (WEP) algorithm for protecting wireless networks. WEP uses RC4 with 40-bit keys, 24-bit initialization vector (IV), and CRC32 to protect against packet forgery. All these choices have proven to be insufficient: key space is too small against current attacks, RC4 key scheduling is insufficient (beginning of the pseudorandom stream should be skipped), IV space is too small and IV reuse makes attacks easier, there is no replay protection, and non-keyed authentication does not protect against bit flipping packet data.

WPA is an intermediate solution for the security issues. It uses Temporal Key Integrity Protocol (TKIP) to replace WEP. TKIP is a compromise on strong security and possibility to use existing hardware. It still uses RC4 for the encryption like WEP, but with per-packet RC4 keys. In addition, it implements replay protection, keyed packet authentication mechanism (Michael MIC).

Keys can be managed using two different mechanisms. WPA can either use an external authentication server (e.g., RADIUS) and EAP just like IEEE 802.1X is using or pre-shared keys without need for additional servers. Wi-Fi calls these "WPA-Enterprise" and "WPA-Personal", respectively. Both mechanisms will generate a master session key for the Authenticator (AP) and Supplicant (client station).

WPA implements a new key handshake (4-Way Handshake and Group Key Handshake) for generating and exchanging data encryption keys between the Authenticator and Supplicant. This handshake is also used to verify that both Authenticator and Supplicant know the master session key. These handshakes are identical regardless of the selected key management mechanism (only the method for generating master session key changes).

IEEE 802.11i / WPA2

The design for parts of IEEE 802.11i that were not included in WPA has finished (May 2004) and this amendment to IEEE 802.11 was approved in June 2004. Wi-Fi Alliance is using the final IEEE 802.11i as a new version of WPA called WPA2. This includes, e.g., support for more robust encryption algorithm (CCMP: AES in Counter mode with CBC-MAC) to replace TKIP and optimizations for handoff (reduced number of messages in initial key handshake, pre-authentication, and PMKSA caching).

wpa_supplicant

wpa_supplicant is an implementation of the WPA Supplicant component, i.e., the part that runs in the client stations. It implements WPA key negotiation with a WPA Authenticator and EAP authentication with Authentication Server. In addition, it controls the roaming and IEEE 802.11 authentication/association of the wlan driver.

wpa_supplicant is designed to be a "daemon" program that runs in the background and acts as the backend component controlling the wireless connection. wpa_supplicant supports separate frontend programs and an example text-based frontend, wpa_cli, is included with wpa_supplicant.

Following steps are used when associating with an AP using WPA:

- wpa_supplicant requests the kernel driver to scan neighboring BSSes
- wpa_supplicant selects a BSS based on its configuration
- wpa_supplicant requests the kernel driver to associate with the chosen BSS
- If WPA-EAP: integrated IEEE 802.1X Supplicant completes EAP

authentication with the authentication server (proxied by the
Authenticator in the AP)
- If WPA-EAP: master key is received from the IEEE 802.1X Supplicant
- If WPA-PSK: wpa_supplicant uses PSK as the master session key
- wpa_supplicant completes WPA 4-Way Handshake and Group Key Handshake
with the Authenticator (AP)
- wpa_supplicant configures encryption keys for unicast and broadcast
- normal data packets can be transmitted and received

Building and installing

In order to be able to build wpa_supplicant, you will first need to select which parts of it will be included. This is done by creating a build time configuration file, .config, in the wpa_supplicant root directory. Configuration options are text lines using following format: CONFIG_<option>=y. Lines starting with # are considered comments and are ignored. See defconfig file for an example configuration and a list of available options and additional notes.

The build time configuration can be used to select only the needed features and limit the binary size and requirements for external libraries. The main configuration parts are the selection of which driver interfaces (e.g., nl80211, wext, ..) and which authentication methods (e.g., EAP-TLS, EAP-PEAP, ..) are included.

Following build time configuration options are used to control IEEE 802.1X/EAPOL and EAP state machines and all EAP methods. Including TLS, PEAP, or TTLS will require linking wpa_supplicant with OpenSSL library for TLS implementation. Alternatively, GnuTLS or the internal TLSv1 implementation can be used for TLS functionality.

```
CONFIG_IEEE8021X_EAPOL=y
CONFIG_EAP_MD5=y
CONFIG_EAP_MSCHAPV2=y
CONFIG_EAP_TLS=y
CONFIG_EAP_PEAP=y
CONFIG_EAP_TTLS=y
```

CONFIG_EAP_GTC=y
CONFIG_EAP_OTP=y
CONFIG_EAP_SIM=y
CONFIG_EAP_AKA=y
CONFIG_EAP_AKA_PRIME=y
CONFIG_EAP_PSK=y
CONFIG_EAP_SAKE=y
CONFIG_EAP_GPSK=y
CONFIG_EAP_PAX=y
CONFIG_EAP_LEAP=y
CONFIG_EAP_IKEV2=y
CONFIG_EAP_PWD=y

Following option can be used to include GSM SIM/USIM interface for GSM/UMTS authentication algorithm (for EAP-SIM/EAP-AKA/EAP-AKA'). This requires pcsc-lite (http://www.linuxnet.com/) for smart card access.

CONFIG_PCSC=y

Following options can be added to .config to select which driver interfaces are included.

CONFIG_DRIVER_NL80211=y
CONFIG_DRIVER_WEXT=y
CONFIG_DRIVER_BSD=y
CONFIG_DRIVER_NDIS=y

Following example includes some more features and driver interfaces that are included in the wpa_supplicant package:

CONFIG_DRIVER_NL80211=y
CONFIG_DRIVER_WEXT=y
CONFIG_DRIVER_BSD=y
CONFIG_DRIVER_NDIS=y
CONFIG_IEEE8021X_EAPOL=y
CONFIG_EAP_MD5=y
CONFIG_EAP_MSCHAPV2=y
CONFIG_EAP_TLS=y
CONFIG_EAP_PEAP=y
CONFIG_EAP_TTLS=y

CONFIG_EAP_GTC=y
CONFIG_EAP_OTP=y
CONFIG_EAP_SIM=y
CONFIG_EAP_AKA=y
CONFIG_EAP_PSK=y
CONFIG_EAP_SAKE=y
CONFIG_EAP_GPSK=y
CONFIG_EAP_PAX=y
CONFIG_EAP_LEAP=y
CONFIG_EAP_IKEV2=y
CONFIG_PCSC=y

EAP-PEAP and EAP-TTLS will automatically include configured EAP methods (MD5, OTP, GTC, MSCHAPV2) for inner authentication selection.

After you have created a configuration file, you can build wpa_supplicant and wpa_cli with 'make' command. You may then install the binaries to a suitable system directory, e.g., /usr/local/bin.

Example commands:

```
# build wpa_supplicant and wpa_cli
make
# install binaries (this may need root privileges)
cp wpa_cli wpa_supplicant /usr/local/bin
```

You will need to make a configuration file, e.g., /etc/wpa_supplicant.conf, with network configuration for the networks you are going to use. Configuration file section below includes explanation fo the configuration file format and includes various examples. Once the configuration is ready, you can test whether the configuration work by first running wpa_supplicant with following command to start it on foreground with debugging enabled:

```
wpa_supplicant -iwlan0 -c/etc/wpa_supplicant.conf -d
```

Assuming everything goes fine, you can start using following command to start wpa_supplicant on background without debugging:

wpa_supplicant -iwlan0 -c/etc/wpa_supplicant.conf -B

Please note that if you included more than one driver interface in the build time configuration (.config), you may need to specify which interface to use by including -D<driver name> option on the command line. See following section for more details on command line options for wpa_supplicant.

Command line options
----

usage:
  wpa_supplicant [-BddfhKLqqtuvW] [-P<pid file>] [-g<global ctrl>] ¥
        [-G<group>] ¥
        -i<ifname> -c<config file> [-C<ctrl>] [-D<driver>] [-p<driver_param>] ¥
        [-b<br_ifname> [-MN -i<ifname> -c<conf> [-C<ctrl>] [-D<driver>] ¥
        [-p<driver_param>] [-b<br_ifname>] [-m<P2P Device config file>] ...

options:
  -b = optional bridge interface name
  -B = run daemon in the background
  -c = Configuration file
  -C = ctrl_interface parameter (only used if -c is not)
  -i = interface name
  -d = increase debugging verbosity (-dd even more)
  -D = driver name (can be multiple drivers: nl80211,wext)
  -f = Log output to default log location (normally /tmp)
  -g = global ctrl_interface
  -G = global ctrl_interface group
  -K = include keys (passwords, etc.) in debug output
  -t = include timestamp in debug messages
  -h = show this help text
  -L = show license (BSD)
  -p = driver parameters
  -P = PID file
  -q = decrease debugging verbosity (-qq even less)
  -u = enable DBus control interface

-v = show version

-W = wait for a control interface monitor before starting

-M = start describing matching interface

-N = start describing new interface

-m = Configuration file for the P2P Device

drivers:

nl80211 = Linux nl80211/cfg80211

wext = Linux wireless extensions (generic)

wired = wpa_supplicant wired Ethernet driver

roboswitch = wpa_supplicant Broadcom switch driver

bsd = BSD 802.11 support (Atheros, etc.)

ndis = Windows NDIS driver

In most common cases, wpa_supplicant is started with

wpa_supplicant -B -c/etc/wpa_supplicant.conf -iwlan0

This makes the process fork into background.

The easiest way to debug problems, and to get debug log for bug reports, is to start wpa_supplicant on foreground with debugging enabled:

wpa_supplicant -c/etc/wpa_supplicant.conf -iwlan0 -d

If the specific driver wrapper is not known beforehand, it is possible to specify multiple comma separated driver wrappers on the command line. wpa_supplicant will use the first driver wrapper that is able to initialize the interface.

wpa_supplicant -Dnl80211,wext -c/etc/wpa_supplicant.conf -iwlan0

wpa_supplicant can control multiple interfaces (radios) either by running one process for each interface separately or by running just one process and list of options at command line. Each interface is separated with -N argument. As an example, following command would start wpa_supplicant for two interfaces:

```
wpa_supplicant ¥
        -c wpa1.conf -i wlan0 -D nl80211 -N ¥
        -c wpa2.conf -i wlan1 -D wext
```

If the interfaces on which wpa_supplicant is to run are not known or do not exist, wpa_supplicant can match an interface when it arrives. Each matched interface is separated with -M argument and the -i argument now allows for pattern matching.

As an example, the following command would start wpa_supplicant for a specific wired interface called lan0, any interface starting with wlan and lastly any other interface. Each match has its own configuration file, and for the wired interface a specific driver has also been given.

```
wpa_supplicant ¥
        -M -c wpa_wired.conf -ilan0 -D wired ¥
        -M -c wpa1.conf -iwlan* ¥
        -M -c wpa2.conf
```

If the interface is added in a Linux bridge (e.g., br0), the bridge interface needs to be configured to wpa_supplicant in addition to the main interface:

```
wpa_supplicant -cw.conf -Dnl80211 -iwlan0 -bbr0
```

Configuration file

wpa_supplicant is configured using a text file that lists all accepted networks and security policies, including pre-shared keys. See example configuration file, wpa_supplicant.conf, for detailed information about the configuration format and supported fields.

Changes to configuration file can be reloaded be sending SIGHUP signal to wpa_supplicant ('killall -HUP wpa_supplicant'). Similarly, reloading can be triggered with 'wpa_cli reconfigure' command.

Configuration file can include one or more network blocks, e.g., one for each used SSID. wpa_supplicant will automatically select the best network based on the order of network blocks in the configuration file, network security level (WPA/WPA2 is preferred), and signal strength.

Example configuration files for some common configurations:

1) WPA-Personal (PSK) as home network and WPA-Enterprise with EAP-TLS as work network

```
# allow frontend (e.g., wpa_cli) to be used by all users in 'wheel' group
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
#
# home network; allow all valid ciphers
network={
	ssid="home"
	scan_ssid=1
	key_mgmt=WPA-PSK
	psk="very secret passphrase"
}
#
# work network; use EAP-TLS with WPA; allow only CCMP and TKIP ciphers
network={
	ssid="work"
	scan_ssid=1
	key_mgmt=WPA-EAP
	pairwise=CCMP TKIP
	group=CCMP TKIP
	eap=TLS
	identity="user@example.com"
	ca_cert="/etc/cert/ca.pem"
	client_cert="/etc/cert/user.pem"
	private_key="/etc/cert/user.prv"
	private_key_passwd="password"
}
```

2) WPA-RADIUS/EAP-PEAP/MSCHAPv2 with RADIUS servers that use old peaplabel

(e.g., Funk Odyssey and SBR, Meetinghouse Aegis, Interlink RAD-Series)

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
network={
        ssid="example"
        scan_ssid=1
        key_mgmt=WPA-EAP
        eap=PEAP
        identity="user@example.com"
        password="foobar"
        ca_cert="/etc/cert/ca.pem"
        phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
}
```

3) EAP-TTLS/EAP-MD5-Challenge configuration with anonymous identity for the unencrypted use. Real identity is sent only within an encrypted TLS tunnel.

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
network={
        ssid="example"
        scan_ssid=1
        key_mgmt=WPA-EAP
        eap=TTLS
        identity="user@example.com"
        anonymous_identity="anonymous@example.com"
        password="foobar"
        ca_cert="/etc/cert/ca.pem"
        phase2="auth=MD5"
}
```

4) IEEE 802.1X (i.e., no WPA) with dynamic WEP keys (require both unicast and broadcast); use EAP-TLS for authentication

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
```

```
network={
        ssid="1x-test"
        scan_ssid=1
        key_mgmt=IEEE8021X
        eap=TLS
        identity="user@example.com"
        ca_cert="/etc/cert/ca.pem"
        client_cert="/etc/cert/user.pem"
        private_key="/etc/cert/user.prv"
        private_key_passwd="password"
        eapol_flags=3
}
```

5) Catch all example that allows more or less all configuration modes. The configuration options are used based on what security policy is used in the selected SSID. This is mostly for testing and is not recommended for normal use.

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
network={
        ssid="example"
        scan_ssid=1
        key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
        pairwise=CCMP TKIP
        group=CCMP TKIP WEP104 WEP40
        psk="very secret passphrase"
        eap=TTLS PEAP TLS
        identity="user@example.com"
        password="foobar"
        ca_cert="/etc/cert/ca.pem"
        client_cert="/etc/cert/user.pem"
        private_key="/etc/cert/user.prv"
        private_key_passwd="password"
        phase1="peaplabel=0"
        ca_cert2="/etc/cert/ca2.pem"
        client_cert2="/etc/cer/user.pem"
        private_key2="/etc/cer/user.prv"
        private_key2_passwd="password"
```

}

6) Authentication for wired Ethernet. This can be used with 'wired' or 'roboswitch' interface (-Dwired or -Droboswitch on command line).

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
ap_scan=0
network={
        key_mgmt=IEEE8021X
        eap=MD5
        identity="user"
        password="password"
        eapol_flags=0
}
```

Certificates

Some EAP authentication methods require use of certificates. EAP-TLS uses both server side and client certificates whereas EAP-PEAP and EAP-TTLS only require the server side certificate. When client certificate is used, a matching private key file has to also be included in configuration. If the private key uses a passphrase, this has to be configured in wpa_supplicant.conf ("private_key_passwd").

wpa_supplicant supports X.509 certificates in PEM and DER formats. User certificate and private key can be included in the same file.

If the user certificate and private key is received in PKCS#12/PFX format, they need to be converted to suitable PEM/DER format for wpa_supplicant. This can be done, e.g., with following commands:

```
# convert client certificate and private key to PEM format
openssl pkcs12 -in example.pfx -out user.pem -clcerts
# convert CA certificate (if included in PFX file) to PEM format
```

openssl pkcs12 -in example.pfx -out ca.pem -cacerts -nokeys

wpa_cli
----

wpa_cli is a text-based frontend program for interacting with wpa_supplicant. It is used to query current status, change configuration, trigger events, and request interactive user input.

wpa_cli can show the current authentication status, selected security mode, dot11 and dot1x MIBs, etc. In addition, it can configure some variables like EAPOL state machine parameters and trigger events like reassociation and IEEE 802.1X logoff/logon. wpa_cli provides a user interface to request authentication information, like username and password, if these are not included in the configuration. This can be used to implement, e.g., one-time-passwords or generic token card authentication where the authentication is based on a challenge-response that uses an external device for generating the response.

The control interface of wpa_supplicant can be configured to allow non-root user access (ctrl_interface_group in the configuration file). This makes it possible to run wpa_cli with a normal user account.

wpa_cli supports two modes: interactive and command line. Both modes share the same command set and the main difference is in interactive mode providing access to unsolicited messages (event messages, username/password requests).

Interactive mode is started when wpa_cli is executed without including the command as a command line parameter. Commands are then entered on the wpa_cli prompt. In command line mode, the same commands are entered as command line arguments for wpa_cli.

Interactive authentication parameters request

When wpa_supplicant need authentication parameters, like username and password, which are not present in the configuration file, it sends a request message to all attached frontend programs, e.g., wpa_cli in interactive mode. wpa_cli shows these requests with "CTRL-REQ-<type>-<id>:<text>" prefix. <type> is IDENTITY, PASSWORD, or OTP (one-time-password). <id> is a unique identifier for the current network. <text> is description of the request. In case of OTP request, it includes the challenge from the authentication server.

The reply to these requests can be given with 'identity', 'password', and 'otp' commands. <id> needs to be copied from the the matching request. 'password' and 'otp' commands can be used regardless of whether the request was for PASSWORD or OTP. The main difference between these two commands is that values given with 'password' are remembered as long as wpa_supplicant is running whereas values given with 'otp' are used only once and then forgotten, i.e., wpa_supplicant will ask frontend for a new value for every use. This can be used to implement one-time-password lists and generic token card -based authentication.

Example request for password and a matching reply:

```
CTRL-REQ-PASSWORD-1:Password needed for SSID foobar
> password 1 mysecretpassword
```

Example request for generic token card challenge-response:

```
CTRL-REQ-OTP-2:Challenge 1235663 needed for SSID foobar
> otp 2 9876
```

wpa_cli commands

```
  status = get current WPA/EAPOL/EAP status
  mib = get MIB variables (dot1x, dot11)
  help = show this usage help
  interface [ifname] = show interfaces/select interface
  level <debug level> = change debug level
  license = show full wpa_cli license
  logoff = IEEE 802.1X EAPOL state machine logoff
```

logon = IEEE 802.1X EAPOL state machine logon

set = set variables (shows list of variables when run without arguments)

pmksa = show PMKSA cache

reassociate = force reassociation

reconfigure = force wpa_supplicant to re-read its configuration file

preauthenticate <BSSID> = force preauthentication

identity <network id> <identity> = configure identity for an SSID

password <network id> <password> = configure password for an SSID

pin <network id> <pin> = configure pin for an SSID

otp <network id> <password> = configure one-time-password for an SSID

passphrase <network id> <passphrase> = configure private key passphrase for an SSID

bssid <network id> <BSSID> = set preferred BSSID for an SSID

list_networks = list configured networks

select_network <network id> = select a network (disable others)

enable_network <network id> = enable a network

disable_network <network id> = disable a network

add_network = add a network

remove_network <network id> = remove a network

set_network <network id> <variable> <value> = set network variables (shows list of variables when run without arguments)

get_network <network id> <variable> = get network variables

save_config = save the current configuration

disconnect = disconnect and wait for reassociate command before connecting

scan = request new BSS scan

scan_results = get latest scan results

get_capability <eap/pairwise/group/key_mgmt/proto/auth_alg> = get capabilies

terminate = terminate wpa_supplicant

quit = exit wpa_cli

wpa_cli command line options

wpa_cli [-p<path to ctrl sockets>] [-i<ifname>] [-hvB] [-a<action file>] ¥
[-P<pid file>] [-g<global ctrl>] [command..]

-h = help (show this usage text)

-v = shown version information

-a = run in daemon mode executing the action file based on events from wpa_supplicant

-B = run a daemon in the background

default path: /var/run/wpa_supplicant
default interface: first interface found in socket path

Using wpa_cli to run external program on connect/disconnect
----

wpa_cli can used to run external programs whenever wpa_supplicant connects or disconnects from a network. This can be used, e.g., to update network configuration and/or trigget DHCP client to update IP addresses, etc.

One wpa_cli process in "action" mode needs to be started for each interface. For example, the following command starts wpa_cli for the default interface (-i can be used to select the interface in case of more than one interface being used at the same time):

```
wpa_cli -a/sbin/wpa_action.sh -B
```

The action file (-a option, /sbin/wpa_action.sh in this example) will be executed whenever wpa_supplicant completes authentication (connect event) or detects disconnection). The action script will be called with two command line arguments: interface name and event (CONNECTED or DISCONNECTED). If the action script needs to get more information about the current network, it can use 'wpa_cli status' to query wpa_supplicant for more information.

Following example can be used as a simple template for an action script:

```
#!/bin/sh

IFNAME=$1
CMD=$2

if [ "$CMD" = "CONNECTED" ]; then
    SSID=`wpa_cli -i$IFNAME status | grep ^ssid= | cut -f2- -d=`
    # configure network, signal DHCP client, etc.
fi
```

```
if [ "$CMD" = "DISCONNECTED" ]; then
	# remove network configuration, if needed
	SSID=
fi
```

Integrating with pcmcia-cs/cardmgr scripts
----

wpa_supplicant needs to be running when using a wireless network with WPA. It can be started either from system startup scripts or from pcmcia-cs/cardmgr scripts (when using PC Cards). WPA handshake must be completed before data frames can be exchanged, so wpa_supplicant should be started before DHCP client.

For example, following small changes to pcmcia-cs scripts can be used to enable WPA support:

Add MODE="Managed" and WPA="y" to the network scheme in /etc/pcmcia/wireless.opts.

Add the following block to the end of 'start' action handler in /etc/pcmcia/wireless:

```
	if [ "$WPA" = "y" -a -x /usr/local/bin/wpa_supplicant ]; then
		/usr/local/bin/wpa_supplicant -B -c/etc/wpa_supplicant.conf ¥
				-i$DEVICE
	fi
```

Add the following block to the end of 'stop' action handler (may need to be separated from other actions) in /etc/pcmcia/wireless:

```
	if [ "$WPA" = "y" -a -x /usr/local/bin/wpa_supplicant ]; then
		killall wpa_supplicant
	fi
```

This will make cardmgr start wpa_supplicant when the card is plugged in.

Dynamic interface add and operation without configuration files
----

wpa_supplicant can be started without any configuration files or network interfaces. When used in this way, a global (i.e., per wpa_supplicant process) control interface is used to add and remove network interfaces. Each network interface can then be configured through a per-network interface control interface. For example, following commands show how to start wpa_supplicant without any network interfaces and then add a network interface and configure a network (SSID):

```
# Start wpa_supplicant in the background
wpa_supplicant -g/var/run/wpa_supplicant-global -B

# Add a new interface (wlan0, no configuration file, driver=nl80211, and
# enable control interface)
wpa_cli -g/var/run/wpa_supplicant-global interface_add wlan0 ¥
        "" nl80211 /var/run/wpa_supplicant

# Configure a network using the newly added network interface:
wpa_cli -iwlan0 add_network
wpa_cli -iwlan0 set_network 0 ssid '"test"'
wpa_cli -iwlan0 set_network 0 key_mgmt WPA-PSK
wpa_cli -iwlan0 set_network 0 psk '"12345678"'
wpa_cli -iwlan0 set_network 0 pairwise TKIP
wpa_cli -iwlan0 set_network 0 group TKIP
wpa_cli -iwlan0 set_network 0 proto WPA
wpa_cli -iwlan0 enable_network 0

# At this point, the new network interface should start trying to associate
# with the WPA-PSK network using SSID test.

# Remove network interface
wpa_cli -g/var/run/wpa_supplicant-global interface_remove wlan0
```

Privilege separation

To minimize the size of code that needs to be run with root privileges (e.g., to control wireless interface operation), wpa_supplicant supports optional privilege separation. If enabled, this separates the privileged operations into a separate process (wpa_priv) while leaving rest of the code (e.g., EAP authentication and WPA handshakes) into an unprivileged process (wpa_supplicant) that can be run as non-root user. Privilege separation restricts the effects of potential software errors by containing the majority of the code in an unprivileged process to avoid full system compromise.

Privilege separation is not enabled by default and it can be enabled by adding CONFIG_PRIVSEP=y to the build configuration (.config). When enabled, the privileged operations (driver wrapper and l2_packet) are linked into a separate daemon program, wpa_priv. The unprivileged program, wpa_supplicant, will be built with a special driver/l2_packet wrappers that communicate with the privileged wpa_priv process to perform the needed operations. wpa_priv can control what privileged are allowed.

wpa_priv needs to be run with network admin privileges (usually, root user). It opens a UNIX domain socket for each interface that is included on the command line; any other interface will be off limits for wpa_supplicant in this kind of configuration. After this, wpa_supplicant can be run as a non-root user (e.g., all standard users on a laptop or as a special non-privileged user account created just for this purpose to limit access to user files even further).

Example configuration:

- create user group for users that are allowed to use wpa_supplicant ('wpapriv' in this example) and assign users that should be able to use wpa_supplicant into that group
- create /var/run/wpa_priv directory for UNIX domain sockets and control user access by setting it accessible only for the wpapriv group:
  mkdir /var/run/wpa_priv
  chown root:wpapriv /var/run/wpa_priv
  chmod 0750 /var/run/wpa_priv
- start wpa_priv as root (e.g., from system startup scripts) with the

enabled interfaces configured on the command line:

wpa_priv -B -P /var/run/wpa_priv.pid nl80211:wlan0

- run wpa_supplicant as non-root with a user that is in wpapriv group:

wpa_supplicant -i ath0 -c wpa_supplicant.conf

wpa_priv does not use the network interface before wpa_supplicant is started, so it is fine to include network interfaces that are not available at the time wpa_priv is started. As an alternative, wpa_priv can be started when an interface is added (hotplug/udev/etc. scripts). wpa_priv can control multiple interface with one process, but it is also possible to run multiple wpa_priv processes at the same time, if desired.

Linux capabilities instead of privileged process

wpa_supplicant performs operations that need special permissions, e.g., to control the network connection. Traditionally this has been achieved by running wpa_supplicant as a privileged process with effective user id 0 (root). Linux capabilities can be used to provide restricted set of capabilities to match the functions needed by wpa_supplicant. The minimum set of capabilities needed for the operations is CAP_NET_ADMIN and CAP_NET_RAW.

setcap(8) can be used to set file capabilities. For example:

sudo setcap cap_net_raw,cap_net_admin+ep wpa_supplicant

Please note that this would give anyone being able to run that wpa_supplicant binary access to the additional capabilities. This can further be limited by file owner/group and mode bits. For example:

sudo chown wpas wpa_supplicant
sudo chmod 0100 wpa_supplicant

This combination of setcap, chown, and chmod commands would allow wpas user to execute wpa_supplicant with additional network admin/raw capabilities.

Common way style of creating a control interface socket in /var/run/wpa_supplicant could not be done by this user, but this directory could be created before starting the wpa_supplicant and set to suitable mode to allow wpa_supplicant to create sockets there. Alternatively, other directory or abstract socket namespace could be used for the control interface.

External requests for radio control

External programs can request wpa_supplicant to not start offchannel operations during other tasks that may need exclusive control of the radio. The RADIO_WORK control interface command can be used for this.

"RADIO_WORK add <name> [freq=<MHz>] [timeout=<seconds>]" command can be used to reserve a slot for radio access. If freq is specified, other radio work items on the same channel may be completed in parallel. Otherwise, all other radio work items are blocked during execution. Timeout is set to 10 seconds by default to avoid blocking wpa_supplicant operations for excessive time. If a longer (or shorter) safety timeout is needed, that can be specified with the optional timeout parameter. This command returns an identifier for the radio work item.

Once the radio work item has been started, "EXT-RADIO-WORK-START <id>" event message is indicated that the external processing can start. Once the operation has been completed, "RADIO_WORK done <id>" is used to indicate that to wpa_supplicant. This allows other radio works to be performed. If this command is forgotten (e.g., due to the external program terminating), wpa_supplicant will time out the radio work item and send "EXT-RADIO-WORK-TIMEOUT <id>" event to indicate that this has happened. "RADIO_WORK done <id>" can also be used to cancel items that have not yet been started.

For example, in wpa_cli interactive mode:

```
> radio_work add test
1
<3>EXT-RADIO-WORK-START 1
```

```
> radio_work show
ext:test@wlan0:0:1:2.487797
> radio_work done 1
OK
> radio_work show


> radio_work done 3
OK
> radio_work show
ext:test freq=2412 timeout=30@wlan0:2412:1:28.583483
<3>EXT-RADIO-WORK-TIMEOUT 2


> radio_work add test2 freq=2412 timeout=60
5
<3>EXT-RADIO-WORK-START 5
> radio_work add test3
6
> radio_work add test4
7
> radio_work show
ext:test2 freq=2412 timeout=60@wlan0:2412:1:9.751844
ext:test3@wlan0:0:0:5.071812
ext:test4@wlan0:0:0:3.143870
> radio_work done 6
OK
> radio_work show
ext:test2 freq=2412 timeout=60@wlan0:2412:1:16.287869
ext:test4@wlan0:0:0:9.679895
> radio_work done 5
OK
<3>EXT-RADIO-WORK-START 7
<3>EXT-RADIO-WORK-TIMEOUT 7
```