

EUROPEAN COMMISSION

Thematic Priority:
SIXTH FRAMEWORK PROGRAM



Priority 2.5.3
INFORMATION SOCIETY TECHNOLOGIES
Unit G3 Embedded Systems



Project Acronym:

SOCRADES

Project Full Title:

**Service-Oriented Cross-layer infRAstructure for
Distributed smart Embedded devices**

Proposal/Contract No: EU FP6 IST-5-034116 IP SOCRADES

Deliverable D5.1.2

Releases Notes for DPWS C stack component

| | |
|---|---------------------|
| Status: | Final |
| Dissemination Level¹: | CONFIDENTIAL |
| Date: | 08.04.2009 |

Organization Name of the Lead Contractor for this Deliverable: **Schneider Electric Industries**

¹ See p. 3 for explanation of Dissemination Levels, as defined in the DoW

Status Description:

| | | | |
|---|---|---|------------|
| Scheduled completion date ² : | M30 (End of February 2008) | Actual completion date ³ : | 08.04.2009 |
| Short document description: | This document contains the Release Notes of the DPWS stack component. | | |
| Author(s) deliverable: | F. Depeisses | Report/deliverable classification: <input checked="" type="checkbox"/> Deliverable <input type="checkbox"/> Three-Monthly Activity Report <input type="checkbox"/> Six-Monthly Activity Report | |
| <input type="checkbox"/> Partner <input type="checkbox"/> Peer reviews <input type="checkbox"/> Contributions | <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Schneider Electric Automation <input type="checkbox"/> <input type="checkbox"/> ABB <input type="checkbox"/> <input type="checkbox"/> APS GmbH <input type="checkbox"/> <input type="checkbox"/> Boliden AB <input type="checkbox"/> <input type="checkbox"/> FlexLink Automation Oy. <input type="checkbox"/> <input type="checkbox"/> Institut f. Automation und Kommunikation e.V. Magdeburg <input type="checkbox"/> <input type="checkbox"/> Kungliga Tekniska Högskolan | <input type="checkbox"/> <input type="checkbox"/> Loughborough University <input type="checkbox"/> <input type="checkbox"/> Luleå University of Technology <input type="checkbox"/> <input type="checkbox"/> Politecnico di Milano <input type="checkbox"/> <input type="checkbox"/> SAP AG <input type="checkbox"/> <input type="checkbox"/> Siemens AG <input type="checkbox"/> <input type="checkbox"/> Tampere University of Technology <input type="checkbox"/> <input type="checkbox"/> Jaguar Cars Ltd. <input type="checkbox"/> <input type="checkbox"/> ARM Ltd. <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Schneider Electric Industries | |
| Peer review approval : | <input checked="" type="checkbox"/> Approved <input type="checkbox"/> Rejected (improve as specified hereunder) | Date: | 06.04.2009 |
| Suggested improvements: | | | |

Version History:

| Version: | Date: | Comments, Changes, Status: | Person(s) ⁴ : |
|----------|------------------------------|----------------------------|--------------------------|
| V0.1 | April 2 nd , 2009 | Initial draft | F. Depeisses |
| V0.2 | April 6 th , 2009 | Final comments | F. Jammes |
| | | | |

² As defined in the DoW

³ Scheduled date for approval

⁴ A list of company short tags can be found in DoW

Table of Contents:

VERSION HISTORY: 2

1. THE DPWS C STACK COMPONENT 4

 1.1. INTRODUCTION 4

 1.2. SOCRADES ARCHITECTURE 5

2. RELEASE NOTES OF THE DPWS C STACK COMPONENT..... 5

 2.1. NEW FEATURES AND ENHANCEMENTS 5

 2.1.1. *Enhancements:* 5

 2.1.2. *New features:*..... 6

 2.1.3. *Build mechanism change*..... 6

 2.2. CHANGES 7

 2.2.1. *API* 7

 2.2.2. *Multi-homed & IPv6 platforms support* 7

 2.3. CONFORMANCE 7

 2.4. DOCUMENTATIONS 7

 2.5. LICENSE 8

 2.6. KNOWN PROBLEMS AND WORKAROUNDS 8

 2.7. LIMITATIONS AND POSSIBLE IMPROVEMENTS 9

 2.8. DOWNLOADING THE DPWS C STACK COMPONENT 9

 2.9. SOCRADES SPECIFIC CONTRIBUTIONS 10

REFERENCES 10

List of Figures:

Figure 1: SOCRADES architecture 5

Dissemination Level:

| | |
|--------------|--|
| PUBLIC | Public |
| PP | Restricted to other programme participants (including the Commission Services) |
| RESTRICTED | Restricted to a group specified by the consortium (including the Commission |
| CONFIDENTIAL | Confidential, only for members of the consortium (including the Commission Services) |

1. The DPWS C stack component

1.1. Introduction

The DPWS C stack component implements the Devices Profile for Web Services specification [1], this component defines a generic software stack to be used for developing devices participating in SOCRADES service-oriented system. This stack is built on top of standard communications protocols (IP v4/v6, TCP, HTTP, UDP/IGMP), usually provided by the device core software (OS, and more precisely a BSD-compatible socket library).

The following figure outlines the DPWS C stack component:

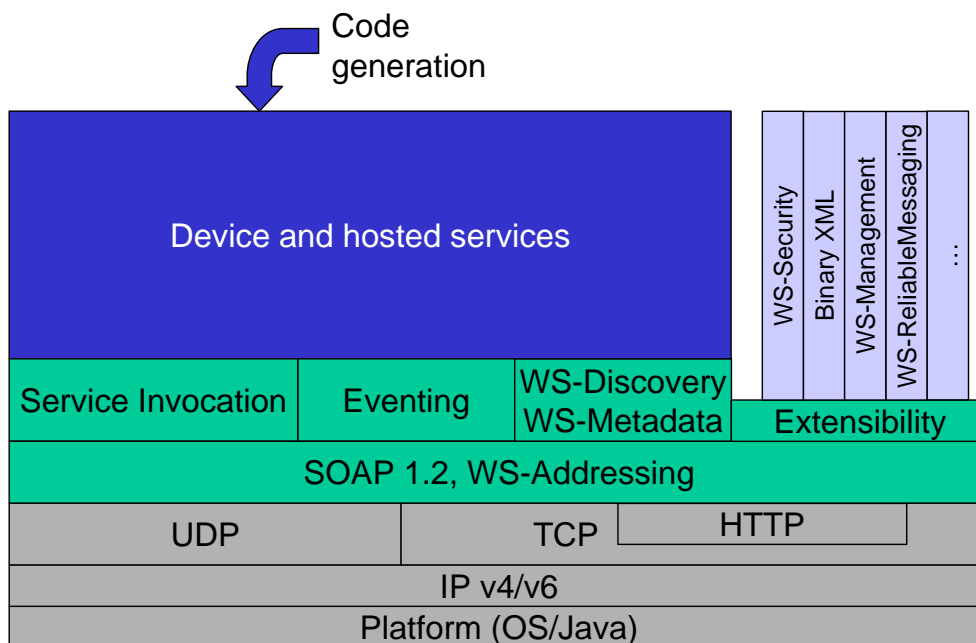


Figure 1. DPWS C stack component

The green layers represent the DPWS C stack component, it provides:

- A service invocation engine based on SOAP 1.2 and WS-Addressing;
- A publish/subscribe engine based on WS-Eventing;
- Ad-hoc discovery and metadata exchange mechanisms based on WS-Discovery, WS-Transfer and WS-MetadataExchange.
- An extensibility mechanism for plugging additional functionalities (not part of the DPWS C stack component), such as:
 - Security through message authentication and encryption
 - Reliable delivery of messages
 - Support for management protocols
 - Support for high-performances messaging (Binary XML)
 - ...

The bottom layers (grey) represent the core software that is expected to be available on each device in order to deploy the DPWS C stack component. The only exception is the HTTP server, which can be provided by the stack if required.

The top layer is the application-specific layer, and consists of user-written and generated code. Code generation tools are provided to assist the user in the development of those device and hosted services.

1.2. SOCRADES architecture

The DPWS C stack component is part of the SOCRADES system architecture denoted in the following overview figure as the orchestration engine box. The figure gives also an overview of the relation-ship to other components which are also part of the deliverables in SOCRADES.

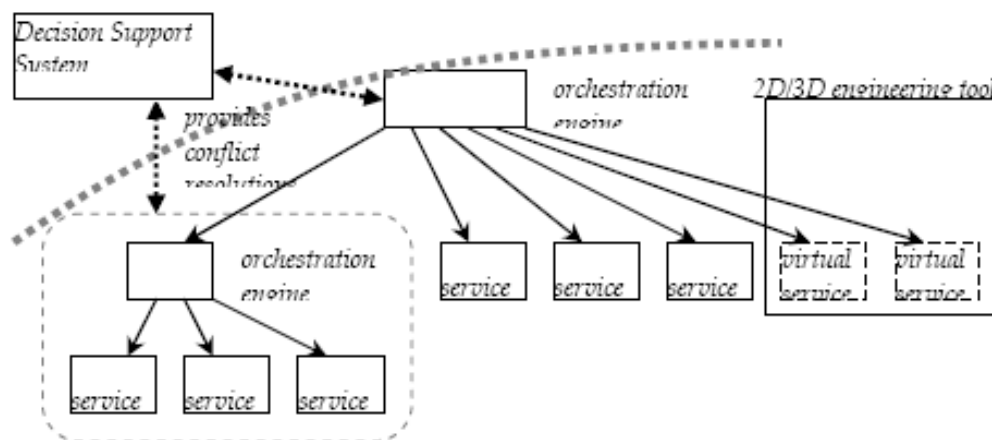


Figure 1: SOCRADES architecture

2. Release Notes of the DPWS C stack component

These release notes cover release information for the **final released version V2.1.0** of the DPWS C stack component. The DPWS C stack component (M30 deliverable) is the second delivery of DPWS C stack component (the first delivery (M24) was the V2.1 M7 version).

2.1. New features and enhancements

This version brings the following main features to the 2.1.0 toolkit:

2.1.1. Enhancements:

- Hot configuration allow to modify devices configuration while they are online and without interruption⁵
- Discovery cache content control with definition of a filter and LRU.
- WS-Eventing subscription manager capacity control,⁶

⁵SOCRADES specific contribution (see §2.9)

⁶ SOCRADES specific contribution (see §2.9)

- Improved device discovery using a single lookup API performing several parallel "PROBE request".
- Full encapsulation of the gSOAP user runtime API.

2.1.2. New features:

- Generic XML processing API (EPX) implemented on top of gSOAP runtime,
- Generic stub & skeleton based on EPX,
- XML file processing facilities,
- WS-Management pluggability support (header management & custom WS-Eventing subscription management plugin interface added),⁷
- Configuration by an XML file,⁸
- WS-Management implementations dispatcher,⁹
- Dynamic deployment framework using the WS-Management protocol (not compliant with the whole specification however),¹⁰
- IPv6 support,
- Multiple interface support,
- External HTTP server pluggability,¹¹
- TCP connection pool (avoids connection overhead and exhaustion),¹²

2.1.3. Build mechanism change

- New code organization & build system.¹³
- API parameters control macros that can be activated with stack build.¹⁴
- Portability layers.
- A full Visual studio 2005 & 2008 build is now available that does not require Cygwin (that previously supplied Bison & Flex) nor GnuWin32 tools anymore, since Bison & Flex output is now provided along with VS projects for 'wsdl2h' and 'soapcpp2'

⁷ SOCRADES specific contribution (see §2.9)

⁸ SOCRADES specific contribution (see §2.9)

⁹ SOCRADES specific contribution (see §2.9)

¹⁰ SOCRADES specific contribution (see §2.9)

¹¹ SOCRADES specific contribution (see §2.9)

¹² SOCRADES specific contribution (see §2.9)

¹³ SOCRADES specific contribution (see §2.9)

¹⁴ SOCRADES specific contribution (see §2.9)

2.2. Changes

2.2.1. API

- A new object can now be configured, the subscription manager. It holds for example a maximum number of subscriptions, and a maximum duration for subscriptions that was moved from the registry to it.
- A `DPWS_INT_MAX_SUBSC_DURATION` attribute has been added to be able to set the maximum duration for subscriptions using an int. Note that `DPWS_STR_MAX_SUBSC_DURATION` cannot be used anymore for reading.
- New configuration parameter on the cache to specify a limit for the number of device proxies in the cache (`DPWS_INT_MAX_DEVICE_PROXIES`).
- A configuration parameter was added (`DPWS_BOOL_BP1_1_COMPATIBILITY` on `DC_TOOLKIT_HANDLE`) in order to turn off WS-Addressing headers use so that to enable WSI Basic Profile clients support, provided the WSDL was written using SOAP 1.1 bindings.
- `dpws_join_network`, added for cable unplugging management, allow the user to send Hello message when a network restoration has been detected.

2.2.2. Multi-homed & IPv6 platforms support

- `dpws_init_ex` modified for multi-homed platforms network interface configuration and DPWS version support. The user can now, for example, supply a sophisticated filter for interface selection while the simplified APIs `dpws_init()` or `dpws_init6()` will continue selecting (without control) one interface supporting a given protocol.
- `dpws_server_init_ex` adapted for listener configuration (most legacy configuration attributes were kept for compatibility reasons).
- `dpws_get_transport_addresses` was added to replace `dpws_get_transport_address` which was kept for backwards compatibility.
- A new WS-Eventing subscription API has been added (`dpws_event_subscribe_ex`) that allow local endpoint handles usage for notification sinks. This API performs automatic EPR selection and hides the multiple interface & protocol complexity.
- `DPWS_STR_HTTP_ADDR` toolkit parameter removed because it interfered with multiple address support.

2.3. Conformance

This toolkit intends to be compliant with at least the minimal requirements of the "Devices Profile for Web Services February 2006" specification from Microsoft Corporation. This version is the one currently supported by the new Windows Vista operating system. It has been tested for interoperability against both Windows Vista final release and the SOA4D DPWS4j Java DPWS Toolkit v1.0.

2.4. Documentations

The **DPWS user guide** (`doc/DPWSCore User Guide.pdf`) should be read first to get an overview of the toolkit features and help on using them.

API reference (doc/api/) brings details API description generated using Doxygen on API header files.

gSOAP documentation (doc/soapdoc2.html) contains the documentation for the gSOAP web service used by this toolkit. Note that the user may still have to refer to it for details about code generation and XML data binding reference even if most DPWS features should be understood by anyone that read the DPWSCore user guide.

All samples are introduced in doc/samples.txt even if some may have a dedicated readme. Anyway, samples contain lots of comments and hints that almost make them tutorials.

In the source package (or on the SVN repository) documents are available for developers that may want to build DPWSCore, for instance to make an embedded target port. doc/dev/DC Build Notes.html will:

- describe the DPWSCore build system,
- give hints for porting DPWS to custom targets (along with the portability layer reference guide.)

2.5. LICENSE

The **DPWSCore runtime** is released under the LGPL license (see file in the package).

The **gSOAP runtime** and the gSOAP compiler (soapcpp2) are released under one of the following three licenses:

- GPL (see file in the package),
- the gSOAP public license (that unlike GPL, allows commercial use. see file in the package),
- the commercial license by Genivia Inc (write to contact@genivia.com or visit <http://genivia.com/Products/gsoap/contract.html>)

This also applies to the code generated by the gSOAP compiler.

The **gSOAP WSDL parser** (wsdl2h) is released under one of the following two licenses:

- GPL (see file in the package),
- the commercial license by Genivia Inc (write to contact@genivia.com or visit <http://genivia.com/Products/gsoap/contract.html>)

This also applies to the code generated by the gSOAP WSDL parser.

NOTE:

Every source file comes with a banner telling under what license it is available, especially because DPWSCore & gSOAP runtimes may be mixed, in a very limited number of files however.

2.6. Known Problems and Workarounds

https://forge.soa4d.org/tracker/?atid=113&group_id=8&func=browse

2.7. Limitations and possible improvements

- The modifications made on the gSOAP compiler only address C code generation. C++ generation is not supported for the time being.
- WSDL is returned only by location (URL) and not yet by reference (WS) or inclusion.
- Client metadata retrieval method is deterministic by section. ThisDevice, ThisModel and Relationship are expected inline while WSDL only by location.
- WS-Eventing : as authorized by the specification the DPWS toolkit only accepts relative time (duration) subscription expiration.

2.8. Downloading the DPWS C stack component

For downloading the package: https://forge.soa4d.org/frs/?group_id=8

2.9. SOCRADES specific contributions

The following points have specifically been developed for SOCRADES and have had major impacts on the DPWS-C stack:

- Integration of the DPWS stack inside the STB Schneider Electric I/O device used in the NOKIA and APS demonstrators. This integration was implying several embedded constraints and had modularity impacts.
- Dynamic deployment, required for integration of the NOKIA demonstrators, was requiring the use of WS-Management.
- The High Level Petri Net interpreter engine, required for the NOKIA demonstrator located in Seligenstadt, had impact on the event delivery mechanisms and on some internal interfaces.

The SOCRADES specific contributions are identified in paragraph 2.1 by a foot note, while most of the other changes are used by SOCRADES but were not specifically implemented by / for SOCRADES.

References

- [1] Devices Profile for Web Services specification, February 2006