

# [MS-DPWSRP]: Devices Profile for Web Services (DPWS): Shared Resource Publishing Data Structure

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

| Date       | Revision History | Revision Class | Comments                                                                     |
|------------|------------------|----------------|------------------------------------------------------------------------------|
| 07/16/2010 | 1.0              | New            | First Release.                                                               |
| 08/27/2010 | 1.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 10/08/2010 | 1.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 11/19/2010 | 1.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 01/07/2011 | 1.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 02/11/2011 | 1.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |

# Contents

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>1 Introduction</b>                                       | <b>4</b>  |
| 1.1 Glossary                                                | 4         |
| 1.2 References                                              | 4         |
| 1.2.1 Normative References                                  | 4         |
| 1.2.2 Informative References                                | 5         |
| 1.3 Overview                                                | 5         |
| 1.4 Relationship to Protocols and Other Structures          | 5         |
| 1.5 Applicability Statement                                 | 5         |
| 1.6 Versioning and Localization                             | 5         |
| 1.7 Vendor-Extensible Fields                                | 5         |
| <b>2 Structures</b>                                         | <b>7</b>  |
| 2.1 The Shell Publishing Data Structure                     | 7         |
| 2.1.1 Namespaces                                            | 7         |
| 2.1.2 Complex Types                                         | 7         |
| 2.1.2.1 pi                                                  | 8         |
| 2.1.2.2 usersFilesDescription                               | 8         |
| 2.1.2.3 o                                                   | 8         |
| 2.1.2.4 il                                                  | 9         |
| 2.1.2.5 i9                                                  |           |
| 2.1.3 Simple Types                                          | 10        |
| 2.1.3.1 serializedType                                      | 10        |
| 2.1.4 Encryption Rules                                      | 10        |
| 2.1.4.1 Data Signing                                        | 10        |
| 2.1.4.2 Data Encoding                                       | 10        |
| 2.1.4.2.1 Alphabet                                          | 10        |
| 2.1.4.2.2 Encoding                                          | 11        |
| <b>3 Structure Examples</b>                                 | <b>13</b> |
| 3.1 Shell Publishing Data Structure Example                 | 13        |
| 3.2 Signed XML Data                                         | 13        |
| 3.3 Base-64-Encoded Shell Publishing Data Structure Example | 14        |
| <b>4 Security</b>                                           | <b>17</b> |
| 4.1 Security Considerations for Implementers                | 17        |
| 4.2 Index of Security Fields                                | 17        |
| <b>5 Appendix A: Product Behavior</b>                       | <b>18</b> |
| <b>6 Change Tracking</b>                                    | <b>19</b> |
| <b>7 Index</b>                                              | <b>20</b> |

# 1 Introduction

The Devices Profile for Web Services (DPWS): Shared Resource Publishing Data Structure describes the Shell Publishing data structure. This data structure is used by the HomeGroup Protocol to advertise shared files and folders in a HomeGroup peer-to-peer network environment.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**alias object**  
**security identifier (SID)**  
**SOAP**  
**Universal Naming Convention (UNC)**  
**XML**  
**XML namespace**  
**XML schema (XSD)**

The following terms are specific to this document:

**item ID list (ID list):** A data structure that refers to a location. An **item ID list** is a multi-segment data structure. Each segment contains content defined by a data source that is responsible for a particular location in the namespace referred to by preceding segments.

**HomeGroup:** A group of one or more computers joined together through the HomeGroup Protocol, which are able to share resources (files, printers, and so on) with each other.

**HomeGroup user:** A user account on the **HomeGroup machine** where files are being shared.

**HomeGroup machine:** The machine where files are being shared, and that creates the Shell Publishing data structure.

**Web Services on Devices (WSD):** A function-discovery protocol used to discover and transfer Shell Publishing data structure in a HomeGroup network environment. Implementation details are specified in [\[DPWS\]](#).

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[DPWS] Chans, S., Conti, D., Schlimmer, J., et al., "Devices Profile for Web Services", February 2006, <http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf>

[MS-SHLLINK] Microsoft Corporation, "[Shell Link \(.LNK\) Binary File Format](#)", July 2010.

[MS-HGRP] Microsoft Corporation, "[HomeGroup Protocol Specification](#)", March 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

[RFC3548] Josefsson, S., Ed., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003, <http://www.ietf.org/rfc/rfc3548.txt>

## 1.2.2 Informative References

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

## 1.3 Overview

This specification extends DPWS [[DPWS](#)] by adding the Shell Publishing data structure. The Shell Publishing data structure describes shared files and folders by each **HomeGroup user** on each **HomeGroup machine** in a **HomeGroup** network environment.

## 1.4 Relationship to Protocols and Other Structures

The Shell Publishing data structure is a data structure format made available to HomeGroup networked environment by a DPWS provider.

| Shell Publishing Extension | This extension    |
|----------------------------|-------------------|
| DPWS                       | Industry standard |
| SOAP                       | Industry standard |

## 1.5 Applicability Statement

Use of the Shell Publishing data structure is suitable when machines in a HomeGroup network environment share files and folders among HomeGroup members.

## 1.6 Versioning and Localization

This document covers versioning issues in the following areas:

- **Supported Transports:** This data structure uses the DPWS provider as the only transport.
- **Protocol Versions:** This data structure is not versioned.
- **Security and Authentication Methods:** This data structure does not support authentication. The data structure is signed using a HomeGroup public key (see [[MS-HGRP](#)] section 3.1.4.5).
- **Localization:** This data structure does not support localization.
- **Capability Negotiation:** This data structure does not support explicit capability negotiation.

## 1.7 Vendor-Extensible Fields

There are no vendor-extensible fields. The **XML schema** of the data structure is not validated, making it possible for vendors to extend the Shell Publishing data structure by adding additional

elements and/or attributes. The extended data will not be interpreted unless consumed by the vendor who added it.

## 2 Structures

### 2.1 The Shell Publishing Data Structure

The Shell Publishing data structure describes a method of publishing and discovering shared files and folders in a HomeGroup configured network environment.

The Shell Publishing data structure MUST be transported using **WSD**. The WSD type MUST be ShellPublishing.

This structure uses SID structures as specified in [\[MS-DTYP\]](#) section 2.4.2.

An individual HomeGroup member MUST publish certain data about his or her shared files and folders as specified in section [2.1.2.1](#), in order to participate in the HomeGroup sharing.

The Shell Publishing XML data structure is defined as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:simpleType name="serializedType">
    <xs:restriction base="xs:string">
      <xs:pattern value="\{[A-Za-z0-9+/*]*\}" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="pi" type="pi" />
</xs:schema>
```

**pi**: A **pi** complex type, as specified in section [2.1.2.1](#). Published items. Serves as an envelope for descriptions of a HomeGroup user's shared files.

#### 2.1.1 Namespaces

| XML Namespace                    | Reference               |
|----------------------------------|-------------------------|
| http://www.w3.org/2001/XMLSchema | <a href="#">[XMLNS]</a> |

#### 2.1.2 Complex Types

The following table summarizes the set of common XML schema complex types defined by this specification.

| Complex Type          | Description                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------|
| pi                    | Published Items. The envelope for the description of shared files and folders.                                           |
| usersFilesDescription | Describes shared files and folders per HomeGroup user, per HomeGroup machine in the HomeGroup.                           |
| o                     | Owner. Describes a HomeGroup user in the HomeGroup machine that is sharing the files and folders on the HomeGroup.       |
| il                    | Items List. Describes a list of items that are being shared by a HomeGroup user in a HomeGroup machine on the HomeGroup. |

| Complex Type | Description                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------------------|
| i            | Item. This type describes a file or folder that is being shared by a HomeGroup user in a HomeGroup machine on the HomeGroup. |

### 2.1.2.1 pi

```
<xs:element name="pi">
  <xs:complexType>
    <xs:element name="usersFilesDescription" type="usersFilesDescription" />
  </xs:complexType>
</xs:element>
```

**usersFilesDescription:** A description of the HomeGroup user's files. Defines the resources shared by a HomeGroup user on a HomeGroup machine.

### 2.1.2.2 usersFilesDescription

```
<xs:element name="usersFilesDescription">
  <xs:complexType>
    <xs:all>
      <xs:element name="o" type="o" />
      <xs:element name="il" type="il" />
    </xs:all>
  </xs:complexType>
</xs:element>
```

**o:** The owner of the shared resource. The owner is typically the HomeGroup user who designates a resource for sharing.

**il:** Contains a sequence of one or more items (the item list).

### 2.1.2.3 o

```
<xs:element name="o">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation>owner information, attributes are user name, alias and
SID</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:attribute name="un" type="xs:string" />
      <xs:attribute name="a" type="xs:string" />
      <xs:attribute name="s" type="xs:string" />
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

**un:** The owner's user name (display name). This is the display name for the HomeGroup user on the HomeGroup machine sharing the files and folders on the HomeGroup.



**a:** The owner's alias. Describes the **alias object** of the HomeGroup user on the HomeGroup machine sharing the files and folders on the HomeGroup. This value MAY be used by the implementation as a hint for the HomeGroup user identity.

**s:** The SID (security identifier) of the account sharing the files on the HomeGroup machine. Describes the security identifier for the HomeGroup user on the HomeGroup machine that is sharing the files and folders on the HomeGroup.

#### 2.1.2.4 il

```
<xs:element name="il">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation>item list</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="i" minOccurs="1" maxOccurs="unbounded" type="i" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**i:** An item in the item list. Contains a description of shared files and folders for the HomeGroup user on the HomeGroup machine.

#### 2.1.2.5 i

```
<xs:element name="i" minOccurs="1" maxOccurs="unbounded">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation>item, sub elements are path (absolute UNC or machine relative),
display name and BASE-64 encoded serialized shell link</xs:documentation>
    </xs:annotation>
    <xs:element name="p" type="xs:anyURI"/>
    <xs:element name="dn" type="xs:string"/>
    <xs:element name="sl" type="serializedType" minOccurs="0"/>
  </xs:complexType>
</xs:element>
```

**i.p:** An absolute **UNC** path or a relative machine path to the shared file or folder. If the path begins with a "\" then it is a machine-relative path. Relative paths are related to the HomeGroup machine where the message originated. The machine name is taken from the WSD Shell Publishing message that is transporting this data structure.

This element is used to access the shared resource if the shell link element **pi.usersFilesDescription.il.i.sl** is not present or if the HomeGroup machine originating the message has changed since the link was created.

**dn:** The display name of the item. The display name is sent so that if the message client implementation uses the display name, it is unnecessary to use additional protocols to retrieve the display name.

**sl:** A base-64-encoded binary stream representing a serialized shell link. The shell link references a file or folder shared by the HomeGroup user on the machine and contains the associated **item ID**

**list.** This information is sent so that if the implementation uses the item ID list, it is unnecessary to use additional protocols to retrieve the ID list. Shell Links are specified in [\[MS-SHLLINK\]](#).

This field uses non-standard base-64 encoding as specified in section [2.1.4.2](#).

### 2.1.3 Simple Types

The following table summarizes the set of common XML schema simple type definitions defined by this specification.

| Simple type    | Description                             |
|----------------|-----------------------------------------|
| serializedType | This is a base-64-encoded binary stream |

#### 2.1.3.1 serializedType

```
<xs:simpleType name="serializedType">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="\{[A-Za-z0-9+/*}\}"/>  
  </xs:restriction>  
</xs:simpleType>
```

**pattern:** This value describes base-64 encoding using the following pattern: `[A-Za-z0-9+/*}\]*`

This field uses non-standard base-64 encoding as specified in section [2.1.4.2](#).

### 2.1.4 Encryption Rules

#### 2.1.4.1 Data Signing

The Shell Publishing data structure MUST be signed using a HomeGroup public key ([\[MS-HGRP\]](#) section 3.1.4.5) prior to being encoded. To create the signature, the data structure is hashed and the hash value is encrypted using HomeGroup public key. This signature is then appended to the data structure.

#### 2.1.4.2 Data Encoding

This data structure MUST be base-64 encoded after being signed and before being transported in a WSD message. The base-64 encoding used by this data structure is a modification on the standard encoding specified by [\[RFC3548\]](#). The alphabet used is the same, but the encoding algorithm is different (see [2.1.4.2.2](#)).

##### 2.1.4.2.1 Alphabet

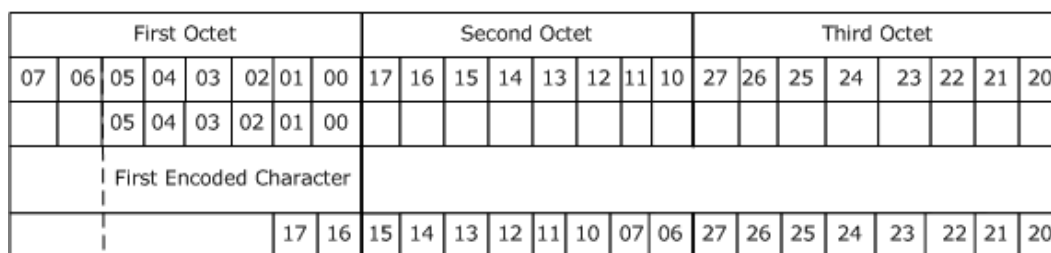
The base-64 alphabet used by this data structure is the following:

| Value | Encoding | Value | Encoding | Value | Encoding | Value | Encoding |
|-------|----------|-------|----------|-------|----------|-------|----------|
| 0     | A        | 17    | R        | 34    | i        | 51    | z        |
| 1     | B        | 18    | S        | 35    | j        | 52    | 0        |
| 2     | C        | 19    | T        | 36    | k        | 53    | 1        |

| Value | Encoding | Value | Encoding | Value | Encoding | Value | Encoding |
|-------|----------|-------|----------|-------|----------|-------|----------|
| 3     | D        | 20    | U        | 37    | l        | 54    | 2        |
| 4     | E        | 21    | V        | 38    | m        | 55    | 3        |
| 5     | F        | 22    | W        | 39    | n        | 56    | 4        |
| 6     | G        | 23    | X        | 40    | o        | 57    | 5        |
| 7     | H        | 24    | Y        | 41    | p        | 58    | 6        |
| 8     | I        | 25    | Z        | 42    | q        | 59    | 7        |
| 9     | J        | 26    | a        | 43    | r        | 60    | 8        |
| 10    | K        | 27    | b        | 44    | s        | 61    | 9        |
| 11    | L        | 28    | c        | 45    | t        | 62    | +        |
| 12    | M        | 29    | d        | 46    | u        | 63    | /        |
| 13    | N        | 30    | e        | 47    | v        |       |          |
| 14    | O        | 31    | f        | 48    | w        |       |          |
| 15    | P        | 32    | g        | 49    | x        |       |          |
| 16    | Q        | 33    | h        | 50    | y        |       |          |

### 2.1.4.2.2 Encoding

The data being encoded is manipulated at the 8-bit chunk (octet) borders. The lowest 6 bits are converted to an appropriate alphabet character. (The value represented by these 6 bits is converted to a corresponding character, shown in the table in [2.1.4.2.1](#)). The remaining 2 bits are combined with the next octet by making them the lowest 2 bits. And the process is repeated, with each step having 2 more extra bits until 6 bits remain, which are then converted to a character without the use of the next octet. The following diagram illustrates this process:



**Figure 1: Data encoding at the 8-bit (octet) level**

The lowest 6 bits of the second octet are converted to the next character, and the remaining 4 bits are moved to be the lowest 4 bits of the next octet.

| First Octet |  |  |  |  |    | Second Octet             |    |    |    |    |    |    |    | Third Octet              |    |    |    |    |    |    |    |    |
|-------------|--|--|--|--|----|--------------------------|----|----|----|----|----|----|----|--------------------------|----|----|----|----|----|----|----|----|
|             |  |  |  |  | 17 | 16                       | 15 | 14 | 13 | 12 | 11 | 10 | 07 | 06                       | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
|             |  |  |  |  |    |                          |    | 13 | 12 | 11 | 10 | 07 | 06 |                          |    |    |    |    |    |    |    |    |
|             |  |  |  |  |    | Second Encoded Character |    |    |    |    |    |    |    |                          |    |    |    |    |    |    |    |    |
|             |  |  |  |  |    |                          |    |    |    | 27 | 26 | 25 | 24 | 23                       | 22 | 21 | 20 | 17 | 16 | 15 | 14 |    |
|             |  |  |  |  |    |                          |    |    |    |    |    |    |    | Third Encoded Character  |    |    |    |    |    |    |    |    |
|             |  |  |  |  |    |                          |    |    |    |    |    |    |    | 27                       | 26 | 25 | 24 | 23 | 22 |    |    |    |
|             |  |  |  |  |    |                          |    |    |    |    |    |    |    | Fourth Encoded Character |    |    |    |    |    |    |    |    |

**Figure 2: Data encoding of the lowest 6 bits and remaining 4 bits**

## 3 Structure Examples

### 3.1 Shell Publishing Data Structure Example

This section contains an example of the Shell Publishing data structure. This example is given raw, before the structure is signed and base-64 encoded. Sections [3.2](#) and [3.3](#) contain examples of signing and base-64 encoding for this particular example.

```
<?xml version="1.0" encoding="UTF-8"?>
<pi>
  <usersFilesDescription>
    <o un="nikola" a="nikola" s="S-1-5-21-2555710863-3024264161-1621211007-1001" />
    <il>
      <i>
        <p>\Users\nikola\AppData\Roaming\Microsoft\Windows\Libraries\Music.library-ms</p>
<sl>MBAAAEAFCAAAAAAAAAADAAAAAYkgAADBgAAAAUOEKgxWpcAF2JNOOnVKHQhdSjjzZlyBmtHAAAAAAAAABAAAAAAAA
AAAAAAAAAAAAAYIAAAHAAAAACAAAAAAAAAAAAACAAAAIEAAAGJAAACAAAAQBAAAAAAAAAAAgAAwFXO10SPxUQtA1QcV
1UFJ1UA4War9GbhxVQwBHRhRXycJl1bh1WaudGXN12Yy92cvZGdcVauR2b3NHXm1mYyFmcpV2cc1Udz12YuwWaiJXyYlX
LtNHAD0AAwAAAKFA8BWN0BLwHivQNEiw03Z8b578MLAAQrAs7rTuznAQAAAAAAAAEAAQMTB1UwEfJ3++RaARpxLAY
M656wCAAAAgCAAAA8BAAAgCAAAAUbQaAsGAvBabAEGAtAAUAMEAAAAAAAAAtAAAAEzUQN1Ok2r3zezgDF55Ehp2pU5qr
AAAAAMAAAAATAAAAAAAAAAAAAAtAAAAEzUQN1cDVuCOUrvPI5pxthzgpbrAAAAAsAAAAALAAAA8//AAAAAAAAAAAA
AAsAwWBUMXc5War9Gbh1CUDxVVzVmczBQTPnmcvN3bmRHIOVgd39mcrBAACAAUAEAAAAAAAAo1OXwJEA4War9GbhBAA6AA
CAQAav7rW7oAnatzFcqCAAawqTDAAGAAAAAAAAAAAAAAAAAAAAuBQaAsGAvBabAEGAAAgFAIFAxAAAAAAAtzCcCBA
BBHcEFGdhBAPAgAAEaw7+q1OLwpW7sAnqAAAAc70AAAAIAAAAAAAAAAAAAAAAAAAQQAHAwBARAEGA0BQYAAAAWAgUA
EDAAAAAAo1OSwJEA1l1bh1WaudGA8AACAQAav7rW7sAnatzEccCAAAAUtDAAAAGAAAAAAAAAAAAAAAAAAAAASBwbAEGAtB
QaA4GAnBAAAYBAYBQMAAAAAAgW70BnQAQTJNkUPN1fxAAAAABACAQAav7rW7sAnatTHcqCAAQuTDAAGAAAAAAAAAA
AAAAAAAAANBQaAMGAYBwbAMHAvBgZAQHAAAAAGAI FAxAAAAAAAtzFcCBAXlmbk92dzBAPAgAAEaw7+q1OLwpW7cBnqAAA
Ao70AAAAIAAAAAAAAAAAAAAAAAAAwVakGAuBAZA8GA3BwcAAAAWAWAEDAAAAAAo1ODyZEAwUSCJVQS5XMAAAQAgAAE
Aw7+q1OXwpW7MInqAAAAgE/AAAAAAAAAAAAAAAAAAAAAAAAAAAAATakGaiBgcAEGAYBQaAUGAZBAAAgBAQCgMAMtHAAgW7M
IngAQTVNVSD5XMuwUSCBgdAgAAEaw7+q1OXwpW7MInqAAAAQ6CBAAAAIAAAAAAAAAAAAAATAAAAAQTUAHHzBQaAMGAuAA
bAkGaiBgcAEGAYBQeA0CatBwCAAAAABwCAgAlBAbAwGazAgMA4CAkBAwGAsAQLAMDA0AQNAgDA0AAAAoBAAAgDAAA
JAAAgA3AAQMTB1UirIWGxLT4M0u8PxmkgZbO3zAAAwAAAAAEBEAAQKDAAAUawHQB+TQDi66kGEiINcAsCMw0ZGA8CR6
wFAAAAAAAAAAAAAAAAAAAAAA0BQMAAAAAAgW7oAnRAQVzVmczBAYAgAAEaw7+e1OVBPw7oAnqAAAAQ60AAAAAAAAAAAA
AAAAAAAAAgNAAAAAQVAMHAlBgcAMHAAAAQAMHAoBQZAwGAsBwMAIDAUAAZAwGAsBALA0CAyAQMAgDaxAwMAAAUUAUAE
AAAAAo1OXwJEA4War9GbhBAA6AACAQAav7rW7oAnatzFcqCAAawqTDAAGAAAAAAAAAAAAAAAAAAAAuBQaAsGAvBab
AEGAAAgFAIFAxAAAAAAAtzCcCBABHcEFGdhBAPAgAAEaw7+q1OLwpW7sAnqAAAAc70AAAAIAAAAAAAAAAAAAAAAA
AQQAHAwBARAEGA0BQYAAAAWAgUAEDAAAAAAo1OSwJEA1l1bh1WaudGA8AACAQAav7rW7sAnatzEccCAAAAUtDAAAAGAAA
AAAAAAAAAAAAAAAAASBwbAEGAtBQaA4GAnBAAAYBAYBQMAAAAAAgW70BnQAQTJNkUPN1fxAAAAABACAQAav7rW7sAnatT
HcqCAAQuTDAAGAAAAAAAAAAAAAAAAAAANBQaAMGAYBwbAMHAvBgZAQHAAAAAGAI FAxAAAAAAAtzFcCBAXlmbk92d
zBAPAgAAEaw7+q1OLwpW7cBnqAAAAo70AAAAIAAAAAAAAAAAAAAAAAAAwVakGAuBAZA8GA3BwcAAAAWAWAEDAAAAAA
o1ODyZEAwUSCJVQS5XMAAAQAgAAEaw7+q1OXwpW7MInqAAAAgE/AAAAAAAAAAAAAAAAAAAAAAAAAAAAATakGaiBgcAEGAYB
QaAUGAZBAAAgBAQCgMAMtHAAgW7MIngAQTVNVSD5XMuwUSCBgdAgAAEaw7+q1OXwpW7MInqAAAAQ6CBAAAAIAAAAAAAAA
AAAAATAAAAAQTUAHHzBQaAMGAuAAbAkGaiBgcAEGAYBQeA0CatBwCAAAAABwCAgAlBAbAwGazAgMA4CAkBAwGAsAQL
AMDA0AQNAgDA0AAAAoBAAAAAAAAQHAAAAACAAAAASAAAzTOLWqWOb10lgo8ypOky4DAAAAAAAAAAgBAAAMAAAAKAAAA
AAAA4War9Gbh1CcjBAAAAAAAAAgDMZXAs83/JVraoUOqv22Mdo1ac+CweHBRgDACU19WTaHT2FAL/9fS1qGKljalTnTt
GnvAs3RwK4AgAVdv1kAAAAAA</sl>
      </i>
    </il>
  </usersFilesDescription>
</pi>
```

### 3.2 Signed XML Data

The following signature has been created based on the Shell Publishing data structure example in section [3.1](#). This signature has been created by hashing the data using the HomeGroup public key (see section [2.1.4.1](#)). This signature can be appended to the Shell Publishing data structure example before base-64 encoding in order to create an authentic message.

```

0x0000:          01 00 00 9d 21 49 20 76
0x0008: ff 72 31 7f 31 5f 57 ef-22 ae 08 92 8e 08 29 5d
0x0028: cd 54 ab 8f 7e 9a 42 ea-a4 c1 03 07 41 38 62 77
0x0048: de 33 cb 83 c2 4f eb b2-cb 10 84 02 8b 22 4f d0
0x0068: 74 e2 04 c6 af 3c 23 8a-5d e3 7c c0 5b b1 84 c2
0x0088: 2c 95 67 aa ff 17 08 7a-48 52 0f 30 2b 6c cd 3d
0x00A8: 3a 24 97 67 0a 68 5d b2-8c 3c a9 d6 90 cf 18 3b
0x00C8: 69 c8 58 de 94 57 e8 39-30 98 0a 79 ac 44 85 02
0x00E8: 21 5e 5e cf 96 24 64 27-59 0a 98 cb 88 68 a5 66
0x0108: 14 1e e6 4a 7d ab e0 15-8e 5b 57 08 3d 7f 0c c3
0x0128: f3 d9 dc 68 95 48 8d 5d-e3 1d 42 3b d0 a1 33 ed
0x0148: f0 30 ea 0e 5c de ca 93-a5 c2 fe a5 72 0c c6 3b
0x0168: c6 aa dd 38 99 dd 44 22-f5 e0 d4 df 74 2a f3 4b
0x0188: 32 c5 55 59 c4 a1 a6 52-3f 9f a2 39 24 33 38 c5
0x01A8: 45 3b 9c f1 24 de be af-41 c0 6d 28 0e 5a 75 4c
0x01C8: 4a 64 5d b0 6d d1 d2-39 2f 1c f3 64 f3 0c 3a
0x01E8: df 9f 00 ec 48 37 01 24-f7 a4 9a

```

### 3.3 Base-64-Encoded Shell Publishing Data Structure Example

This section contains a signed, base-64-encoded version of the Shell Publishing data structure example provided in section 3.1. The base-64 encoding has been applied to the structure after appending the signature given in section 3.2.

```

0x0000: TxAAAwzP4lGbgYXZyNXav5WPiEjLwICI
0x0020: 152YvRWaudWPiUFVG1COi8jPNoAPwlmP
0x0040: NoAIgwTdzVmczZUasV2cEV2cjJXawRXa
0x0060: v5mPNoAIgACI88GI15Wpi4War9GbhJCI
0x0080: h1jIul2avxWYiAyc9IyUtETL10iMx0iM
0x00A0: 1UTN3EDM4YzMtMDMyQjM2QTM2ETLxYjM
0x00C0: xITMxADM30SMwATMiAyL+OgCgACIgwTa
0x00E0: s5TDKACIgACIgwTa+0gCgACIgACIgACP
0x0100: w5DXVNXZyNHXul2avxWYcFEcwRUY0FGX
0x0120: S9WYt1mbnxVTPnmcvN3bmRHXX1mbk92d
0x0140: zxFTpJmchJXa1NHXNV3cpNmLs1mYyFmc
0x0160: 51SbzxzLw5TDKACIgACIgACI8MHb+0kQ
0x0180: BFUQFFkRDFUQBFUQBFUQBFUQBFUQBFUQ
0x01A0: rdWQBRkQnFUQBFUQBFUQBFUQBFUQBFUQ
0x01C0: K50TP5mVLhUUoR2UqpmeaxWeC1EdIFUQ
0x01E0: BFUQBFUQCFUQBFUQBFUQBFUQBFUQBFUQ
0x0200: BFUQZ1UQBFUQIFUQBF0QBFUQBFUQBFUQ
0x0220: BFUQBFUQjFUQBFUSFFUQBFUQBFUQBFUQ
0x0240: BFUQRJUQBFUQBFUQBFUQBFUQBFUQBFUQ
0x0260: wMFU4VVU0FUMRNmVxU1RKFTVBRzVhJXO
0x0280: HJGa4ZVU3JESShmUY11YKFjYoFzVhVHZ
0x02A0: Hh1TsJTW51jMjZnWHR2YkZVY1J1MiNjT
0x02C0: IhVTs1WW5ZUbjBnVyM2YxUFZ6xmMZV3d
0x02E0: XFWaKhVW5xGWMRnTIFERPFUQBFUQBFUQ
0x0300: LZUQ4I0VO9mQmHSpZXUOVUa390MahjY
0x0320: 1cDONxUQBFUyF0c3IHV1pnbBFVQBFUQ
0x0340: BFUQFVUQBFUUNR1QxU1dFZmSzsYKSFwQ
0x0360: SBHeMFUWNZTN2M3VDFUQBFUQBFUQBFUQ
0x0380: CFUQBFUQBFUQBFUQBFUQBFUQBFUQBFUQ
0x03A0: HFEdbFUVB1URBFUQBFUQBFUQBFUQBFUQ
0x03C0: 6VVUOx2TrJjczoxZ6dGRGVTFhGcyAXV

```



0x0B40: BFUY0p3QjNkQBjKQINWRGdEZoJUQQF0Z  
0x0B60: BFURBd3NrEXMPx0dwd1NzFkbxFUQBF0Y  
0x0B80: 3ATQBFUQB1UQBFUQBFBF0QBFUQBFBF0Q  
0x0BA0: BFUQBFBVUBFESBdnQBjVQFdUQwIUUZFUQ  
0x0BC0: BF0VBdWVBVERBFUQBFBF0QvFzTTdnSFFUS  
0x0BE0: xIGaxcVY1R2RBhTQBNUQRFUQ2djcxDzc  
0x0C00: B5WY0pWRjF3QBFUQBHVHVEFUQBFB0ZBFUQ  
0x0C20: BFUQBFBF0QBFUQBFBF0QBFUQBTJ0diFUR  
0x0C40: HFEdcFVYBRzRB5mQBFUQZJUQZJUUNFUQ  
0x0C60: BFUQBd2V3AjQuFVQRR1SotWVQ5EbmhXQ  
0x0C80: BFUQCF0QBFBVQBZ3Nyd1NzFkhhRHVINWc  
0x0CA0: DFUQBFBVdURUQBFBF0QnFUQBFBF0QBFUQ  
0x0CC0: BFUQBFBF0QBFUQB5kQRFWQNdUQ5J0diFUT  
0x0CE0: IFkdCdmWBFFSFBF0QdUQJZUQ4FUQBFBF0Q  
0x0D00: BFUY0pnRjNkQBhFbtJ2a5IDZ6JUQQF0Z  
0x0D20: BFURBd3NrEXMPx0dwd1NjJkbxFUQBF0b  
0x0D40: 3ATQBFUQB1UQBFUQBFBF0QBFUQBFBF0Q  
0x0D60: BFUQBdnVBt2RBVnQBpVQ4cUQzI0djFUQ  
0x0D80: BF0VBF0VBVERBFUQBFBF0QvFzTElnWFF0d  
0x0DA0: VN1QKZVUTVDWNFUQBFBF0QnFUQFF0d3sSc  
0x0DC0: x8EW3B3V30USuFXQBFBF0QnV0LBFUQBFBF0T  
0x0DE0: BFUQBFBF0QBFUQBFBF0QBFUQBFBF0QF0a  
0x0E00: HFUaCd2YBV0RB1nQRFWQVdUQ6JUQBFBF0Z  
0x0E20: CFUUDdWTB1EdIFUQnd1NN1kbnFUUUZ1T  
0x0E40: WNFR1gVT1dXVTNkQnRWQnFUQFF0d3sSc  
0x0E60: x8EW3B3V30USuFXQBFBF0QRZzQCFUQBFBF0S  
0x0E80: BFUQBFBF0QBFUQBFBF0QBRVQBFBF0UUUFUV  
0x0EA0: IFkeCFVYB10RBVXQBjWQrdUQpJ0ZjFUR  
0x0EC0: HFUeCFVZBBzQBRnQ3NWQBFBF0QBJ0djF0Z  
0x0EE0: HFEbCFkYBd3RBpXQn1UQOMUQRJUQiF0d  
0x0F00: HF0cBFFTB1ERBBTQR5UQnRUQwEUQBFB0b  
0x0F20: CFUQBFBF0QBFUUIFUQBFB0QBFUQBFBF0QTFUQ  
0x0F40: BFkeU9ETXF3dPJWmww2ZvhTew90a5RDR  
0x0F60: BFUQBFBF0QBFUQBdmQBFBF0QnFUQBFB0SXFUQ  
0x0F80: BFUQBFBF0QBRzVhJXOHJGaxM0YqJUQBFBF0Q  
0x0FA0: BFUQBdGZNpFWBNHOz8iSWJXYvV1TxZ1M  
0x0FC0: y0EZPFTYjtyQ3VGSCJ3ZEF0QVFTOXRvY  
0x0FE0: IR1MGFETvkjZTFTcHtEbqFGb05EVuRfd  
0x1000: H5mdBN3MSd3S0E0ZBZFZ2FzaBFUQBFBF0Q  
0x1020: 88ycs5TDKACIgACIgwzLp5TDKACIgACP  
0x1040: vkGb+0gCgACPvU3c1J3cG1GblNHR1N3Y  
0x1060: y1Gc012bu5TDKwzLwlmPAEAAA0ZIJBid  
0x1080: /LXM/FzXX9uIuiqkOiQKd1Mvr+ofaKk6  
0x10A0: kG8AHEEOidn3zs8gC/06yuMEEKwii8E0  
0x10C0: 0JOBG/KPjoYXjzHwbFLhCzS1np6/Xgge  
0x10E0: IJ1DwsCbN3jOkc5ZKgWXyyIPpaNkPjxO  
0x1100: phMWeT5VondMYqQesSUhCEiXe9s1kQ2J  
0x1120: ZpAmLjIalaGFeYuS9tK4V44WXhQP/xww  
0x1140: znN3oVJSN214dI0OQH6MtDPMq7AXer8k  
0x1160: lKs/lKHDGvjxq2NOZ2NRiUP4U/NdqM/S  
0x1180: yUcVZRcomK1PfKaOkMDOFX0OcGPJe77r  
0x11A0: BBcbo4gW1xkSk1Fsw2W0SnzLcMPZzzgO  
0x11C0: f/JAsj0NBQy9kqJ



## 4 Security

### 4.1 Security Considerations for Implementers

The Shell Publishing data structure relies on HomeGroup key signing to validate authenticity of the data.

### 4.2 Index of Security Fields

| Security parameter   | Section                 |
|----------------------|-------------------------|
| HomeGroup public key | <a href="#">2.1.4.1</a> |

## 5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows® 7 operating system
- Windows® Home Server 2011 server software

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

## 6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

## 7 Index

### A

[Applicability](#) 5

### B

[Base-64-encoded shell publishing data structure example](#) 14

### C

[Change tracking](#) 19

Complex types

[i](#) 9

[il](#) 9

[o](#) 8

[overview](#) 7

[pi](#) 8

[usersFilesDescription](#) 8

### D

Data

[encoding encryption rules](#) 10

[signing encryption rules](#) 10

[Data structures Shell Publishing](#) 7

Details

[complex types](#) 7

data

[encoding encryption rules](#) 10

[signing encryption rules](#) 10

[i complex type](#) 9

[il complex type](#) 9

[namespaces](#) 7

[o complex type](#) 8

[pi complex type](#) 8

[serializedType simple type](#) 10

[Shell Publishing data structure](#) 7

[simple types](#) 10

[usersFilesDescription complex type](#) 8

### E

Encryption rules

data

[encoding](#) 10

[signing](#) 10

Examples

[base-64-encoded shell publishing data structure](#) 14

[shell publishing data structure](#) 13

[signed XML data](#) 13

### F

[Fields - security index](#) 17

[Fields - vendor-extensible](#) 5

### G

[Glossary](#) 4

### I

[i complex type](#) 9

[il complex type](#) 9

[Implementer - security considerations](#) 17

[Index of security fields](#) 17

[Informative references](#) 5

[Introduction](#) 4

### L

[Localization](#) 5

### N

[Namespaces](#) 7

[Normative references](#) 4

### O

[o complex type](#) 8

[Overview \(synopsis\)](#) 5

### P

[pi complex type](#) 8

[Product behavior](#) 18

### R

References

[informative](#) 5

[normative](#) 4

[Relationship to protocols and other structures](#) 5

### S

Security

[field index](#) 17

[implementer considerations](#) 17

[serializedType simple type](#) 10

[Shell Publishing data structure](#) 7

[Shell publishing data structure example](#) 13

[Signed XML data example](#) 13

Simple types

[overview](#) 10

[serializedType](#) 10

Structures

[Shell Publishing](#) 7

[simple types](#) 10

### T

[Tracking changes](#) 19

Types

[complex](#) 7

[simple](#) 10

## **U**

[usersFilesDescription\\_complex\\_type](#) 8

## **V**

[Vendor-extensible fields](#) 5

[Versioning](#) 5