

## [MS-PAC]: Privilege Attribute Certificate Data Structure

This topic lists the Errata found in [MS-PAC] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.



Errata are subject to the same terms as the Open Specifications documentation referenced.

Errata below are for Protocol Document Version [V22.0 - 2021/06/25](#).

Errata Published*	Description
2022/01/25	<p>In section <b>2.3 PACTYPE</b>, added code snippet.</p> <p>Changed from: The PACTYPE structure is defined as follows: &lt;BitTable&gt;</p> <p>Changed to: The PACTYPE structure is defined as follows. typedef struct _PACTYPE {     ULONG cBuffers;     ULONG Version;     PAC_INFO_BUFFER Buffers[1]; } PACTYPE, *PPACTYPE;</p> <p>The format of the PACTYPE structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.4 PAC_INFO_BUFFER</b>, added code snippet.</p> <p>Changed from: Following the PACTYPE (section 2.3) structure is an array of PAC_INFO_BUFFER structures... the ordering of the buffers MUST NOT change, or signature verification of the PAC contents will fail. &lt;BitTable&gt;</p> <p>Changed to: Following the PACTYPE (section 2.3) structure is an array of PAC_INFO_BUFFER structures... the ordering of the buffers MUST NOT change, or signature verification of the PAC contents will fail.</p> <p>The PAC_INFO_BUFFER structure is defined as follows. typedef struct _PAC_INFO_BUFFER {     ULONG ulType;     ULONG cbBufferSize;     ULONG64 Offset; } PAC_INFO_BUFFER, *PPAC_INFO_BUFFER;</p> <p>The format of the PAC_INFO_BUFFER structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.5 KERB_VALIDATION_INFO</b>, adjusted code snippet.</p> <p>Changed from: typedef struct _KERB_VALIDATION_INFO {     ... } KERB_VALIDATION_INFO;</p> <p>Changed to: typedef struct _KERB_VALIDATION_INFO {     ... } KERB_VALIDATION_INFO, *PKERB_VALIDATION_INFO;</p> <p>In section <b>2.6.1 PAC_CREDENTIAL_INFO</b>, added code snippet.</p>

Errata Published*	Description
	<p>Changed from: The PAC_CREDENTIAL_INFO structure serves as the header for the credential information. ... The PAC credentials buffer is included only when PKINIT [RFC4556] is used. Therefore, the AS reply key is derived based on PKINIT. &lt;BitTable&gt;</p> <p>Changed to: The PAC_CREDENTIAL_INFO structure serves as the header for the credential information. ... The PAC credentials buffer is included only when PKINIT [RFC4556] is used. Therefore, the AS reply key is derived based on PKINIT. The PAC_CREDENTIAL_INFO structure is defined as follows. typedef struct _PAC_CREDENTIAL_INFO {     ULONG Version;     ULONG EncryptionType;     UCHAR SerializedData[1]; } PAC_CREDENTIAL_INFO, *PPAC_CREDENTIAL_INFO; The format of the PAC_CREDENTIAL_INFO structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.7 PAC_CLIENT_INFO</b>, added code snippet.</p> <p>Changed from: The PAC_CLIENT_INFO structure is a variable length buffer of the PAC that contains the client's name and authentication time. ... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000A. &lt;BitTable&gt;</p> <p>Changed to: The PAC_CLIENT_INFO structure is a variable length buffer of the PAC that contains the client's name and authentication time. ... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000A.</p> <p>The PAC_CLIENT_INFO structure is defined as follows. typedef struct _PAC_CLIENT_INFO {     FILETIME ClientId;     USHORT NameLength;     WCHAR Name[1]; } PAC_CLIENT_INFO, *PPAC_CLIENT_INFO; The format of the PAC_CLIENT_INFO structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.8 PAC_SIGNATURE_DATA</b>, added code snippet.</p> <p>Changed from: Two PAC_SIGNATURE_DATA structures are appended to the PAC... No other protocol can use these PAC signatures.</p> <p>The format of the PAC_SIGNATURE_DATA structures is defined as follows: &lt;BitTable&gt;</p> <p>Changed to: Two PAC_SIGNATURE_DATA structures are appended to the PAC... No other protocol can use these PAC signatures.</p> <p>The PAC_SIGNATURE_DATA structure is defined as follows. typedef struct _PAC_SIGNATURE_DATA {     ULONG SignatureType;     UCHAR Signature[ANYSIZE_ARRAY]; // size is from the PAC_INFO_BUFFER - sizeof(ULONG) } PAC_SIGNATURE_DATA, *PPAC_SIGNATURE_DATA; The format of the PAC_SIGNATURE_DATA structure is defined as follows. &lt;BitTable&gt;</p>

Errata Published*	Description
	<p>In section <b>2.10 UPN_DNS_INFO</b>, added code snippet.</p> <p>Changed from: The UPN_DNS_INFO structure contains the client's UPN,... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000C.&lt;19&gt; &lt;BitTable&gt;</p> <p>Changed to: The UPN_DNS_INFO structure contains the client's UPN,... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000C.&lt;19&gt;</p> <p>The UPN_DNS_INFO structure is defined as follows. typedef struct _UPN_DNS_INFO {     USHORT UpnLength;     USHORT UpnOffset;     USHORT DnsDomainNameLength;     USHORT DnsDomainNameOffset;     ULONG Flags; } UPN_DNS_INFO, *PUPN_DNS_INFO; The format of the UPN_DNS_INFO structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.11 PAC_CLIENT_CLAIMS_INFO</b>, added code snippet.</p> <p>Changed from: The PAC_CLIENT_CLAIMS_INFO structure is a variable length buffer of the PAC... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000D. &lt;BitTable&gt;</p> <p>Changed to: The PAC_CLIENT_CLAIMS_INFO structure is a variable length buffer of the PAC... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000D.</p> <p>The PAC_CLIENT_CLAIMS_INFO structure is defined as follows. typedef struct _PAC_CLIENT_CLAIMS_INFO {     PCLAIMS_SET_METADATA Claims; } PAC_CLIENT_CLAIMS_INFO, *PPAC_CLIENT_CLAIMS_INFO; The format of the PAC_CLIENT_CLAIMS_INFO structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.13 PAC_DEVICE_CLAIMS_INFO</b>, added code snippet.</p> <p>Changed from: The PAC_DEVICE_CLAIMS_INFO structure is a variable length buffer of the PAC... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000F. &lt;BitTable&gt;</p> <p>Changed to: The PAC_DEVICE_CLAIMS_INFO structure is a variable length buffer of the PAC... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x0000000F.</p> <p>The PAC_DEVICE_CLAIMS_INFO structure is defined as follows. typedef struct _PAC_DEVICE_CLAIMS_INFO {     PCLAIMS_SET_METADATA Claims; } PAC_DEVICE_CLAIMS_INFO, *PPAC_DEVICE_CLAIMS_INFO; The format of the PAC_DEVICE_CLAIMS_INFO structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.14 PAC_ATTRIBUTES_INFO</b>, added code snippet.</p> <p>Changed from:</p>

Errata Published*	Description
	<p>The PAC_ATTRIBUTES_INFO structure is a variable length buffer of the PAC... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x00000011. &lt;BitTable&gt;</p> <p>Changed to: The PAC_ATTRIBUTES_INFO structure is a variable length buffer of the PAC... The ulType field of the corresponding PAC_INFO_BUFFER is set to 0x00000011.</p> <p>The PAC_ATTRIBUTES_INFO structure is defined as follows. typedef struct _PAC_ATTRIBUTES_INFO {     ULONG FlagsLength;          // specified in bits     ULONG Flags[ANYSIZE_ARRAY]; } PAC_ATTRIBUTES_INFO, *PPAC_ATTRIBUTES_INFO;</p> <p>The format of the PAC_ATTRIBUTES_INFO structure is as follows. &lt;BitTable&gt;</p> <p>In section <b>2.16 Formal MIDL Definition</b>, added and adjusted code snippets.</p> <p>Changed from:</p> <pre> ... typedef struct _PAC_CLIENT_INFO {     FILETIME ClientId;     USHORT NameLength;     WCHAR Name[1]; } PAC_CLIENT_INFO, *PPAC_CLIENT_INFO;  typedef struct _S4U_DELEGATION_INFO {     RPC_UNICODE_STRING S4U2proxyTarget;     ULONG TransitedListSize;     [size_is( TransitedListSize )]     PRPC_UNICODE_STRING S4UTransitedServices; } S4U_DELEGATION_INFO, *PS4U_DELEGATION_INFO; ... typedef struct _PAC_DEVICE_CLAIMS_INFO {     PCLAIMS_SET_METADATA Claims; } PAC_DEVICE_CLAIMS_INFO, *PPAC_DEVICE_CLAIMS_INFO; </pre> <p>Changed to:</p> <pre> ... typedef struct _PAC_CLIENT_INFO {     FILETIME ClientId;     USHORT NameLength;     WCHAR Name[1]; } PAC_CLIENT_INFO, *PPAC_CLIENT_INFO;  typedef struct _PAC_SIGNATURE_DATA {     ULONG SignatureType;     UCHAR Signature[ANYSIZE_ARRAY];  // size is from the PAC_INFO_BUFFER - sizeof(ULONG) } PAC_SIGNATURE_DATA, *PPAC_SIGNATURE_DATA;  typedef struct _S4U_DELEGATION_INFO {  // Constrained Delegation Information section 2.9     RPC_UNICODE_STRING S4U2proxyTarget;     ULONG TransitedListSize;     [size_is( TransitedListSize )]     PRPC_UNICODE_STRING S4UTransitedServices; } S4U_DELEGATION_INFO, *PS4U_DELEGATION_INFO; ... typedef struct _PAC_DEVICE_CLAIMS_INFO {     PCLAIMS_SET_METADATA Claims; } PAC_DEVICE_CLAIMS_INFO, *PPAC_DEVICE_CLAIMS_INFO; </pre>

Errata Published*	Description														
	<pre>typedef struct _PAC_ATTRIBUTES_INFO {     ULONG FlagsLength;           // specified in bits     ULONG Flags[ANYSIZE_ARRAY]; } PAC_ATTRIBUTES_INFO, *PPAC_ATTRIBUTES_INFO;</pre>														
2022/01/25	<p>In section <b>2.4 PAC_INFO_BUFFER</b>, added new <b>ulType</b> values for PAC Attributes and PAC Requestor.</p> <p>Changed from:</p> <table border="1" data-bbox="397 535 1412 745"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>...</td> <td></td> </tr> <tr> <td>0x00000010</td> <td>Ticket checksum (section 2.8). PAC structures SHOULD NOT contain more than one buffer of this type. Additional ticket checksum buffers MUST be ignored.&lt;7&gt;</td> </tr> </tbody> </table> <p>Changed to:</p> <table border="1" data-bbox="397 787 1412 1144"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0x00000010</td> <td>Ticket checksum (section 2.8). PAC structures SHOULD NOT contain more than one buffer of this type. Additional ticket checksum buffers MUST be ignored.&lt;7&gt;</td> </tr> <tr> <td>0x00000011</td> <td>PAC Attributes indicates that the buffer contains attribute bits for the PAC (section 2.14). PAC structures SHOULD NOT contain more than one buffer of this type. Additional attribute buffers MUST be ignored.&lt;8&gt;</td> </tr> <tr> <td>0x00000012</td> <td>PAC Requestor indicates that the buffer contains the SID of principal that requested the PAC (section 2.15). PAC structures MUST contain one buffer of this type.&lt;9&gt;</td> </tr> </tbody> </table> <p>&lt;8&gt;The PAC Attributes value is not supported in Windows Server 2008 Service Pack 1 and earlier.          &lt;9&gt;The PAC Requestor value is not supported in Windows Server 2008 Service Pack 1 and earlier.</p>	Value	Meaning	...		0x00000010	Ticket checksum (section 2.8). PAC structures SHOULD NOT contain more than one buffer of this type. Additional ticket checksum buffers MUST be ignored.<7>	Value	Meaning	0x00000010	Ticket checksum (section 2.8). PAC structures SHOULD NOT contain more than one buffer of this type. Additional ticket checksum buffers MUST be ignored.<7>	0x00000011	PAC Attributes indicates that the buffer contains attribute bits for the PAC (section 2.14). PAC structures SHOULD NOT contain more than one buffer of this type. Additional attribute buffers MUST be ignored.<8>	0x00000012	PAC Requestor indicates that the buffer contains the SID of principal that requested the PAC (section 2.15). PAC structures MUST contain one buffer of this type.<9>
Value	Meaning														
...															
0x00000010	Ticket checksum (section 2.8). PAC structures SHOULD NOT contain more than one buffer of this type. Additional ticket checksum buffers MUST be ignored.<7>														
Value	Meaning														
0x00000010	Ticket checksum (section 2.8). PAC structures SHOULD NOT contain more than one buffer of this type. Additional ticket checksum buffers MUST be ignored.<7>														
0x00000011	PAC Attributes indicates that the buffer contains attribute bits for the PAC (section 2.14). PAC structures SHOULD NOT contain more than one buffer of this type. Additional attribute buffers MUST be ignored.<8>														
0x00000012	PAC Requestor indicates that the buffer contains the SID of principal that requested the PAC (section 2.15). PAC structures MUST contain one buffer of this type.<9>														
2022/01/25	<p>In section <b>2.14 PAC_ATTRIBUTES_INFO</b>, clarified field definitions.</p> <p>Changed from:</p> <p><b>FlagsLength (4 bytes):</b> An unsigned 32-bit integer in little-endian format that describes the length, in bits, of the Flags field.</p> <p><b>Flags (variable):</b> an array of 32-bit unsigned integers in little-endian format that contains flag bits describing the PAC. The following bits are currently defined:          &lt;Table&gt;          All other bits are zero and MUST be ignored on receipt.</p> <p>Changed to:</p> <p><b>FlagsLength (4 bytes):</b> An unsigned 32-bit integer in little-endian format that describes the length, in bits, of the Flags field as the count of flag bits not including padding.</p> <p><b>Flags (variable):</b> an array of 32-bit unsigned integers in little-endian format that contains the flag bits describing the PAC. Each 32-bit unsigned integer contains up to 32 defined bits. All bits are flag bits padded to a unit multiple of 32 bits of storage (a 32-bit integer). Undefined padding bits are zero and MUST be ignored on receipt. The following bits are currently defined.          &lt;Table&gt;</p>														

<b>Errata Published*</b>	<b>Description</b>
2021/11/11	<p>The following sections were changed or added. Please see the <a href="#">diff document</a> for the details.</p> <p>In section 2.10 UPN_DNS_INFO, added four new fields and a flag to the UPN_DNS_INFO structure.</p> <p>In section 2.14 PAC_ATTRIBUTES_INFO, added section.</p> <p>In section 2.15 PAC_REQUESTOR, added section.</p>