

[MS-WUSP]: Windows Update Services: Client-Server Protocol

This topic lists the Errata found in [MS-WMF] 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 [V28.0 – 2018/09/12](#).

Errata Published*	Description
2020/02/17	<p>In Section 3.1.5.1, Self-Update, the text has been updated to clarify the steps taken by clients which use the self-update procedure. Additionally, information has been provided as to how to initiate a license request for the Microsoft-signed redistributable files needed by Windows 7, 8, and 8.1 clients for a third-party implementation of the self-update process and more details provided regarding the processing of the files.</p> <p>Changed from:</p> <p>At the start of the protocol, the client SHOULD<31> check if it needs to self-update. Client/server communications can fail if the server does not expose the self-update content directory, as specified in this section.</p> <p>As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. The client issues HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory. The directory MUST include the wuident.cab file, which is implementation-specific.<32></p> <p><31> Section 3.1.5.1: Only Windows 8.1 and earlier Windows versions use this self-update mechanism.</p> <p><32> Section 3.1.5.1: Windows Server Update Services includes a version of wuident.cab that is compatible with clients running Windows 8.1 and earlier. A current copy of this wuident.cab file is available at https://go.microsoft.com/fwlink/?linkid=2111079.</p> <p>Changed to:</p> <p>At the start of the protocol, a client that implements the self-update protocol SHOULD<31> check if it needs to self-update. Client/server communications can fail if the server does not expose the self-update content directory, as specified in this section.</p> <p>As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. Clients that support the self-update protocol issue HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory to support these clients. The files that the client accesses in the self-update directory, and the way in which the client uses them, are implementation-specific.<32></p> <p><31> Section 3.1.5.1: Only Windows 8.1 and earlier Windows versions use this self-update mechanism.</p> <p><32> Windows 8.1 and earlier Windows versions perform self-update using the contents of the self-update content directory. Since these versions of Windows require that these files be Microsoft-signed, the files cannot be modified. For information on how to receive a redistributable copy of these files, contact dochelp@microsoft.com.</p>

Errata Published*	Description
	<p>Windows 10 does not perform self-update and does not access the files in the self-update content directory.</p> <p>For reference, here is a summary of how Windows 8.1 and earlier Windows versions perform self-update.</p> <ol style="list-style-type: none"> 1. The client retrieves the file wuident.cab from the self-update content directory and verifies that it is signed by Microsoft. If the wuident.cab file is not present or is not signed by Microsoft, the client stops self-update and does not further interoperate with the server. 2. The client retrieves the file wuident.txt from the wuident.cab file. 3. The client compares the version number of the current version of the Windows Update client files to the versions listed in the [OS], [OS2], and [OS3] sections of the wuident.txt file, and finds the matching line in the file. 4. If the matching line in wuident.txt contains the directive "/SKIP", then the client is already up-to-date and self-update is complete. 5. If the matching line in wuident contains the directive "/UNSUPPORTED", then the client cannot self-update and does not further interoperate with the server. 6. If the matching line in wuident contains neither "/SKIP" nor "/UNSUPPORTED", then it specifies a subdirectory. 7. The client compares the architecture of the current version of windows to the list of architectures in the [ARCH] section of the wuident.txt file and retrieves a subdirectory name from the matching line. 8. The client compares the primary language of the current version of windows to the list of languages in the [LANG] section of the wuident.txt file and retrieves a subdirectory name from the matching line. 9. The client assembles the full subdirectory path from the three subdirectory names it has retrieved, using the format specified in the StructureKeyEx, StructureKeyEx2, or StructureKeyEx3 lines in the [SelfClientUpdate] section of the wuident.txt file. 10. The client accesses the resulting self-directory within the self-update content directory. This directory will contain a set of several files containing the appropriate Windows Update client binaries for the current version, architecture, and language of Windows. 11. The client verifies that these files are Microsoft-signed, unpacks the contents into a secure directory, and executes them to update the client binaries. 12. If the installation succeeds, then self-update is complete. 13. If the installation fails, then the client cannot self-update and does not further interoperate with the server.
2019/12/09	<p>In Section 3.1.1.1, Populating the Data Model, the text was updated to include the OSUpgrade attribute in the description of the concatenated strings of the XMLFragment component of both the Core and Extended metadata entries.</p> <p>Changed from:</p> <p>Core: There is exactly one "Core" entry created from the revisions metadata.</p> <p>RevisionID: References the entry in the revision table for the revisions metadata.</p> <p>FragmentType: Core.</p> <p>Locale: NULL.</p> <p>XmlFragment: MUST be derived from the original metadata by:</p> <ul style="list-style-type: none"> • Collecting the following XmlNode nodes: • The XmlNode identified by XPATH /Update/UpdateIdentity • The XmlNode identified by XPATH /Update/Properties, all attributes removed except: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, and EulaID <p>...</p>

Errata Published*	Description
	<p>Extended: There is exactly one "Extended" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Extended. Locale: NULL. XmlFragment: MUST be derived from the original metadata by concatenating the following strings together after removing all XML namespace definitions from each string (the result of which is not well-formed XML):</p> <ul style="list-style-type: none"> • The XmlNode identified by XPATH /Update/Properties, with the following attributes removed: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, EulaID, PublicationState, PublisherID, CreationDate, IsPublic, LegacyName, DetectoidType... <p>Changed to:</p> <p>Core: There is exactly one "Core" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Core. Locale: NULL. XmlFragment: MUST be derived from the original metadata by:</p> <ul style="list-style-type: none"> • Collecting the following XmlNodes: • The XmlNode identified by XPATH /Update/UpdateIdentity • The XmlNode identified by XPATH /Update/Properties, all attributes removed except: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, OSUpgrade, and EulaID <p>...</p> <p>Extended: There is exactly one "Extended" entry created from the revisions metadata. RevisionID: References the entry in the revision table for the revisions metadata. FragmentType: Extended. Locale: NULL. XmlFragment: MUST be derived from the original metadata by concatenating the following strings together after removing all XML namespace definitions from each string (the result of which is not well-formed XML):</p> <ul style="list-style-type: none"> • The XmlNode identified by XPATH /Update/Properties, with the following attributes removed: UpdateType, ExplicitlyDeployable, AutoSelectOnWebSites, EulaID, PublicationState, PublisherID, CreationDate, IsPublic, LegacyName, DetectoidType, OSUpgrade. <p>...</p>
2019/11/25	<p>In Section 3.1.5.1, Self-Update, the text was changed to provide information as to where the update files could be accessed.</p> <p>Changed from:</p> <p>At the start of the protocol, the client MUST check if it needs to self-update. Client/server communications fail if the server does not expose the self-update content directory, as specified in this section.</p> <p>As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. The client issues HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory.</p>

Errata Published*	Description				
	<p>Changed to:</p> <p>At the start of the protocol, the client SHOULD<31> check if it needs to self-update. Client/server communications can fail if the server does not expose the self-update content directory, as specified in this section.</p> <p>As specified in section 2, the server implementation MUST expose the self-update content directory as a virtual directory. The client issues HTTP GET requests (as specified in [RFC2616] section 9.3) to obtain files from the self-update content directory; therefore, the server MUST support HTTP requests on this virtual directory. The directory MUST include the wuident.cab file, which is implementation-specific<32>.</p> <p><31>Only Windows 8.1 and earlier Windows versions use this self-update mechanism.</p> <p><32>Windows Server Update Services includes a version of wuident.cab that is compatible with clients running Windows 8.1 and earlier. A current copy of this wuident.cab file is available at https://go.microsoft.com/fwlink/?linkid=2111079.</p>				
2019/11/11	<p>In Section 2.2.2.3.1, ReportEventBatch, an event was added to the Install Events section of the EventID table.</p> <p>Added:</p> <table border="1" data-bbox="399 1031 1417 1129"> <tr> <td data-bbox="399 1031 761 1129">AGENT_INSTALLING_STARTED</td> <td data-bbox="761 1031 837 1129">181</td> <td data-bbox="837 1031 1032 1129">User initiated installation started</td> <td data-bbox="1032 1031 1417 1129">Installation Started:Windows has started installing the following update: %1</td> </tr> </table>	AGENT_INSTALLING_STARTED	181	User initiated installation started	Installation Started:Windows has started installing the following update: %1
AGENT_INSTALLING_STARTED	181	User initiated installation started	Installation Started:Windows has started installing the following update: %1		
2019/06/24	<p>In this document, the properties of the LastChangeTime element of the Deployment complex type have been updated to indicate that it is mandatory.</p> <p>In Section 2.2.2.2.4, SyncUpdates, the minOccurs value for the LastChangeTime element of the Deployment complex type has been changed from "0" to "1".</p> <p>Changed from:</p> <p>...</p> <pre data-bbox="435 1457 1300 1885"> <s:complexType name="Deployment"> <s:sequence> <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:int" /> <s:element minOccurs="1" maxOccurs="1" name="Action" type="s1:DeploymentAction" /> <s:element minOccurs="0" maxOccurs="1" name="Deadline" type="s:string" /> <s:element minOccurs="1" maxOccurs="1" name="IsAssigned" type="s:boolean" /> <s:element minOccurs="0" maxOccurs="1" name="LastChangeTime" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="DownloadPriority" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="HardwareIds" type="s1:ArrayOfString" /> <s:element minOccurs="0" maxOccurs="1" name="AutoSelect" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="AutoDownload" </pre>				

Errata Published*	Description
	<pre> type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="SupersedenceBehavior" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="FlagBitmask" type="s:string" /> </s:sequence> </s:complexType> ... LastChangeTime: Specifies when the deployment was created, in the syntax specified for s:date (as specified in [XMLSCHEMA2] section 3.2.9). ... Changed to: ... <s:complexType name="Deployment"> <s:sequence> <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:int" /> <s:element minOccurs="1" maxOccurs="1" name="Action" type="s1:DeploymentAction" /> <s:element minOccurs="0" maxOccurs="1" name="Deadline" type="s:string" /> <s:element minOccurs="1" maxOccurs="1" name="IsAssigned" type="s:boolean" /> <s:element minOccurs="1" maxOccurs="1" name="LastChangeTime" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="DownloadPriority" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="HardwareIds" type="s1:ArrayOfString" /> <s:element minOccurs="0" maxOccurs="1" name="AutoSelect" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="AutoDownload" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="SupersedenceBehavior" type="s:string" /> <s:element minOccurs="0" maxOccurs="1" name="FlagBitmask" type="s:string" /> </s:sequence> </s:complexType> ... LastChangeTime: Specifies when the deployment was created, in the syntax specified for s:date (as specified in [XMLSCHEMA2] section 3.2.9). This element MUST be present. ... </pre>

Errata Published*	Description
	<p>In Section 6.2, Client Web Service WSDL, changed from:</p> <p>...</p> <pre data-bbox="435 394 1403 443"><s:element minOccurs="0" maxOccurs="1" name="LastChangeTime" type="s:string" /></pre> <p>...</p> <p>Changed to:</p> <p>...</p> <pre data-bbox="435 646 1403 695"><s:element minOccurs="1" maxOccurs="1" name="LastChangeTime" type="s:string" /></pre> <p>...</p>
2019/04/29	<p>In Section 3.1.5.7, SyncUpdates, and Section 3.1.5.12, SyncPrinterCatalog, a note has been added to clarify that only one revision can be sent for a given update.</p> <p>In Section 3.1.5.7, SyncUpdates, changed from:</p> <p>...</p> <p>Xml: The revision's associated "core" metadata (FragmentType = "Core"). SyncUpdatesResponse.OutOfScopeRevisionIDs: Populated with the IDs of revision that are in the CachedRevisions list that are not in the NeededRevisions list.</p> <p>...</p> <p>Changed to:</p> <p>...</p> <p>Xml: The revision's associated "core" metadata (FragmentType = "Core"). Note: The server implementation MUST send no more than one Revision for a given Update. It is recommended that the implementation SHOULD, in the event of multiple matches, select only the latest Revision (the one with the highest revision number). SyncUpdatesResponse.OutOfScopeRevisionIDs: Populated with the IDs of revision that are in the CachedRevisions list that are not in the NeededRevisions list.</p> <p>...</p> <p>In Section 3.1.5.12, SyncPrinterCatalog, changed from:</p> <p>...</p> <p>Xml: The "core" metadata (FragmentType = "Core") associated with the revision. SyncPrinterCatalogResponse.OutOfScopeRevisionIDs: Populated with the revision's IDs in the CachedRevisions list that are not in the NeededRevisions list.</p> <p>...</p> <p>Changed to:</p> <p>...</p> <p>Xml: The "core" metadata (FragmentType = "Core") associated with the revision.</p>

Errata Published*	Description
	<p>Note: The server implementation MUST send no more than one Revision for a given Update. It is recommended that the implementation SHOULD, in the event of multiple matches, select only the latest Revision (the one with the highest revision number).</p> <p>SyncPrinterCatalogResponse.OutOfScopeRevisionIDs: Populated with the revision's IDs in the CachedRevisions list that are not in the NeededRevisions list.</p> <p>...</p>

*Date format: YYYY/MM/DD