[MS-ASPROV]:

Exchange ActiveSync: Provisioning Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
12/3/2008	1.0.0	Major	Initial Release.
3/4/2009	1.0.1	Editorial	Revised and edited technical content.
4/10/2009	2.0.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0.0	Major	Revised and edited for technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	3.1.0	None	Version 3.1.0 Release
5/5/2010	4.0.0	Major	Updated and revised the technical content.
8/4/2010	5.0	Major	Significantly changed the technical content.
11/3/2010	5.1	Minor	Clarified the meaning of the technical content.
3/18/2011	6.0	Major	Significantly changed the technical content.
8/5/2011	6.1	Minor	Clarified the meaning of the technical content.
10/7/2011	6.2	Minor	Clarified the meaning of the technical content.
1/20/2012	7.0	Major	Significantly changed the technical content.
4/27/2012	7.1	Minor	Clarified the meaning of the technical content.
7/16/2012	8.0	Major	Significantly changed the technical content.
10/8/2012	9.0	Major	Significantly changed the technical content.
2/11/2013	10.0	Major	Significantly changed the technical content.
7/26/2013	11.0	Major	Significantly changed the technical content.
11/18/2013	11.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	11.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	12.0	Major	Significantly changed the technical content.
7/31/2014	12.1	Minor	Clarified the meaning of the technical content.
10/30/2014	13.0	Major	Significantly changed the technical content.
5/26/2015	14.0	Major	Significantly changed the technical content.
6/30/2015	15.0	Major	Significantly changed the technical content.
9/14/2015	16.0	Major	Significantly changed the technical content.
6/9/2016	17.0	Major	Significantly changed the technical content.
2/28/2017	18.0	Major	Significantly changed the technical content.
4/18/2017	18.0	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
9/19/2017	19.0	Major	Significantly changed the technical content.
12/12/2017	19.1	Minor	Clarified the meaning of the technical content.
7/24/2018	20.0	Major	Significantly changed the technical content.
10/1/2018	21.0	Major	Significantly changed the technical content.
12/11/2018	21.1	Minor	Clarified the meaning of the technical content.
4/29/2022	22.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Exchange ActiveSync: Provisioning Protocol describes an **XML**-based format used by servers that support the ActiveSync protocol to communicate security policy settings to client devices.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

cabinet (.cab) file: A single file that stores multiple compressed files to facilitate storage or transmission.

encrypted message: An Internet email message that is in the format described by [RFC5751] and uses the EnvelopedData CMS content type described in [RFC3852], or the Message object that represents such a message.

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

plain text: Text that does not have markup. See also plain text message body.

policy key: A stored value that represents the state of a policy or setting.

remote wipe: Functionality that is implemented on a client, initiated by policy or a request from a server, that requires the client to delete all data and settings related to the referenced protocol.

Short Message Service (SMS): A communications protocol that is designed for sending text messages between mobile phones.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Wireless Application Protocol (WAP) Binary XML (WBXML): A compact binary representation of **XML** that is designed to reduce the transmission size of XML documents over narrowband communication channels.

XML: The Extensible Markup Language, as described in [XML1.0].

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic

syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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. We will assist you in finding the relevant information.

[MS-ASCMD] Microsoft Corporation, "Exchange ActiveSync: Command Reference Protocol".

[MS-ASDTYPE] Microsoft Corporation, "Exchange ActiveSync: Data Types".

[MS-ASHTTP] Microsoft Corporation, "Exchange ActiveSync: HTTP Protocol".

[MS-ASWBXML] Microsoft Corporation, "Exchange ActiveSync: WAP Binary XML (WBXML) Algorithm".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/rfc/rfc2119.html

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, https://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, https://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

1.2.2 Informative References

[MS-ASAIRS] Microsoft Corporation, "Exchange ActiveSync: AirSyncBase Namespace Protocol".

[MS-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

[MSDN-MSPROVDTDFormat] Microsoft Corporation, "MSPROV DTD Format", http://msdn.microsoft.com/en-us/library/bb737266.aspx

1.3 Overview

This protocol consists of an **XML schema** that defines the elements that are necessary for an ActiveSync device to specify its capabilities and **permissions**.

1.4 Relationship to Other Protocols

This protocol describes the XML format that is used by the **Provision** command. The structure of ActiveSync command requests and responses is specified in [MS-ASHTTP].

All simple data types in this document conform to the data type definitions specified in [MS-ASDTYPE].

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol describes a set of elements for use in communicating device capabilities and security requirements between a client and a server. This protocol is applicable to clients that conform to server security requirements, and to servers that implement security requirements and capability criteria for client devices.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol consists of a series of XML elements contained in request or response messages that is associated with the **Provision** command between a client and server.

The encoded XML block containing the command and parameter elements is transmitted in either the request body of a request, or in the response body of a response.

All Provision command messages are encoded as **Wireless Application Protocol (WAP) Binary XML (WBXML)**, as specified in [MS-ASWBXML].

2.2 Message Syntax

The XML schema for the Provision namespace is described in section 6.

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
None	**Provision**	
folderhierarchy	**FolderHierarchy**	[MS-ASCMD] sections 2.2.1.3, 2.2.1.4, 2.2.1.5, 2.2.1.6, 2.2.1.8
settings	**Settings**	[MS-ASCMD] section 2.2.1.18
xs	**http://www.w3.org/2001/XMLSchema**	[XMLSCHEMA1]

2.2.2 Elements

The following table summarizes the set of common XML schema element definitions that are defined or used by this specification. XML schema elements that are specific to a particular command are described in the context of its associated command.

Element name	Description
AccountOnlyRemoteWipe (section 2.2.2.1)	Specifies either an account only **remote wipe** directive from the server or a client's confirmation of an account only remote wipe directive.
AllowBluetooth (section 2.2.2.2)	Whether Bluetooth and hands-free profiles are allowed on the device.
AllowBrowser (section 2.2.2.3)	Whether the device allows the use of a web browser.
AllowCamera (section 2.2.2.4)	Whether the device allows the use of the built-in camera.
AllowConsumerEmail (section 2.2.2.5)	Whether the device allows the use of personal email.

Element name	Description
AllowDesktopSync (section 2.2.2.6)	Whether the device allows synchronization with Desktop ActiveSync.
AllowHTMLEmail (section 2.2.2.7)	Whether the device uses **HTML**-formatted email.
AllowInternetSharing (section 2.2.2.8)	Whether the device allows the use of Internet Sharing.
AllowIrDA (section 2.2.2.9)	Whether the device allows the use of IrDA (infrared) connections.
AllowPOPIMAPEmail (section 2.2.2.10)	Whether the device allows access to POP/IMAP email.
AllowRemoteDesktop (section 2.2.2.11)	Whether the device allows the use of Remote Desktop.
AllowSimpleDevicePassword (section 2.2.2.12)	Whether the device allows simple passwords.
AllowSMIMEEncryptionAlgorithmNegotiation (section 2.2.2.13)	Whether the device can negotiate the encryption algorithm to be used for signing.
AllowSMIMESoftCerts (section 2.2.2.14)	Whether the device uses soft certificates to sign outgoing messages.
AllowStorageCard (section 2.2.2.15)	Whether the device allows the use of the storage card.
AllowTextMessaging (section 2.2.2.16)	Whether the device allows **Short Message Service (SMS)**/text messaging.
AllowUnsignedApplications (section 2.2.2.17)	Whether the device allows unsigned applications to execute.
AllowUnsignedInstallationPackages (section 2.2.2.18)	Whether the device allows unsigned **cabinet (.cab) files** to be installed.
AllowWiFi (section 2.2.2.19)	Whether the device allows the use of Wi-Fi connections.
AlphanumericDevicePasswordRequired (section 2.2.2.20)	Indicates whether a client device requires an alphanumeric password.
ApplicationName (section 2.2.2.21)	The name of an in-ROM application (.exe file) that is not approved for execution.
ApprovedApplicationList (section 2.2.2.22)	A list of in-RAM applications that are approved for execution.
AttachmentsEnabled (section 2.2.2.23)	Indicates whether email attachments are enabled.
Data (section 2.2.2.24)	The settings for a policy.
DevicePasswordEnabled (section 2.2.2.25)	Indicates whether a client device requires a password.
DevicePasswordExpiration (section 2.2.2.26)	Whether the password expires after the specified number of days, as determined by the policy.
DevicePasswordHistory (section 2.2.2.27)	The minimum number of previously used passwords the client device stores to prevent reuse.
EASProvisionDoc (section 2.2.2.28)	The collection of security settings for device provisioning.
Hash (section 2.2.2.29)	The SHA-1 hash of an in-memory application that is approved for execution.
MaxAttachmentSize (section 2.2.2.30)	The maximum attachment size, as determined by the security policy.
MaxCalendarAgeFilter (section 2.2.2.31)	The maximum number of calendar days that can be synchronized.
MaxDevicePasswordFailedAttempts (section 2.2.2.32)	The number of password failures that are permitted before the device is wiped.
MaxEmailAgeFilter (section 2.2.2.33)	The email age limit for synchronization.

Element name	Description
MaxEmailBodyTruncationSize (section 2.2.2.34)	The truncation size for **plain text**–formatted email messages.
MaxEmailHTMLBodyTruncationSize (section 2.2.2.35)	The truncation size for HTML-formatted email messages.
MaxInactivityTimeDeviceLock (section 2.2.2.36)	The number of seconds of inactivity before the device locks itself.
MinDevicePasswordComplexCharacters (section 2.2.2.37)	The minimum number of complex characters (numbers and symbols) contained within the password.
MinDevicePasswordLength (section 2.2.2.38)	The minimum device password length that the user can enter.
PasswordRecoveryEnabled (section 2.2.2.39)	Indicates whether to enable a recovery password to be sent to the server by using the **Settings** command.
Policies (section 2.2.2.40)	A collection of security policies.
Policy (section 2.2.2.41)	A policy.
PolicyKey (section 2.2.2.42)	Used by the server to mark the state of policy settings on the client.
PolicyType (section 2.2.2.43)	Specifies the format in which the policy settings are to be provided.
Provision (section 2.2.2.44)	The capabilities and permissions for the device.
RemoteWipe (section 2.2.2.45)	Specifies either a remote wipe directive from the server or a client's confirmation of a remote wipe directive.
RequireDeviceEncryption (section 2.2.2.46)	Whether the device uses encryption.
RequireEncryptedSMIMEMessages (section 2.2.2.47)	Whether the device is required to send **encrypted messages**.
RequireEncryptionSMIMEAlgorithm (section 2.2.2.48)	The algorithm to be used when encrypting a message.
RequireManualSyncWhenRoaming (section 2.2.2.49)	Whether the device requires manual synchronization when the device is roaming.
RequireSignedSMIMEAlgorithm (section 2.2.2.50)	The algorithm to be used when signing a message.
RequireSignedSMIMEMessages (section 2.2.2.51)	Whether the device is required to send signed S/MIME messages.
RequireStorageCardEncryption (section 2.2.2.52)	Indicates whether the device has to encrypt content that is stored on the storage card.
settings:DeviceInformation (section 2.2.2.53)	Specifies the settings for the device in an initial Provisioning request.
Status (section 2.2.2.54)	Indicates success or failure of specific parts of a command.
UnapprovedInROMApplicationList (section 2.2.2.55)	A list of in-ROM applications that are not approved for execution.

2.2.2.1 AccountOnlyRemoteWipe

The **AccountOnlyRemoteWipe** element is an optional **container** ([MS-ASDTYPE] section 2.2) element that specifies either an account only remote wipe directive from the server or a client's confirmation of a server's account only remote wipe directive.

A server response MUST NOT include any child elements in the **AccountOnlyRemoteWipe** element.

The **AccountOnlyRemoteWipe** element is sent in a command request only in response to an account only remote wipe directive from the server.

The **AccountOnlyRemoteWipe** element has the following child element in a command request:

- **Status** (section 2.2.2.54.3): One element of this type is required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	
14.0	
14.1	
16.0	
16.1	Yes

2.2.2.2 AllowBluetooth

The **AllowBluetooth** element is an optional child element of type **unsignedByte** ([MS-ASDTYPE] section 2.8) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the use of Bluetooth on the device.

The **AllowBluetooth** element cannot have child elements.

Valid values for **AllowBluetooth** are listed in the following table.

Value	Meaning
0	Disable Bluetooth.
1	Disable Bluetooth, but allow the configuration of hands-free profiles.
2	Allow Bluetooth.

This element SHOULD be ignored if the client does not support Bluetooth.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.3 AllowBrowser

The **AllowBrowser** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of a web browser.

The **AllowBrowser** element cannot have child elements.

Valid values for **AllowBrowser** are listed in the following table.

Value	Meaning
0	Do not allow the use of a web browser.
1	Allow the use of a web browser.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.4 AllowCamera

The **AllowCamera** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of the built-in camera.

The **AllowCamera** element cannot have child elements.

Valid values for **AllowCamera** are listed in the following table.

Value	Meaning
0	Use of the camera is not allowed.
1	Use of the camera is allowed.

This element SHOULD be ignored if the client does not have a camera and no camera can be attached to the device.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.5 AllowConsumerEmail

The **AllowConsumerEmail** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the user to configure a personal email account.

The **AllowConsumerEmail** element cannot have child elements.

Valid values for **AllowConsumerEmail** are listed in the following table.

Value	Meaning
0	Do not allow the user to configure a personal email account.
1	Allow the user to configure a personal email account.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.6 AllowDesktopSync

The **AllowDesktopSync** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows synchronization with Desktop ActiveSync.

The **AllowDesktopSync** element cannot have child elements.

Valid values for **AllowDesktopSync** are listed in the following table.

Value	Meaning
0	Do not allow Desktop ActiveSync.
1	Allow Desktop ActiveSync.

This element SHOULD be ignored if the client does not support connecting to a personal computer.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes

Protocol version	Element support
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.7 AllowHTMLEmail

The **AllowHTMLEmail** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the client uses HTML-formatted email.

The **AllowHTMLEmail** element cannot have child elements.

Valid values for **AllowHTMLEmail** are listed in the following table.

Value	Meaning
0	HTML-formatted email is not allowed.
1	HTML-formatted email is allowed.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.8 AllowInternetSharing

The **AllowInternetSharing** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of Internet Sharing.

The **AllowInternetSharing** element cannot have child elements.

Valid values for **AllowInternetSharing** are listed in the following table.

Value	Meaning
0	Do not allow the use of Internet Sharing.
1	Allow the use of Internet Sharing.

This element SHOULD be ignored if the client does not support sharing its internet connection with other devices.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.9 AllowIrDA

The **AllowIrDA** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of IrDA (infrared) connections.

The **AllowIrDA** element cannot have child elements.

Valid values for **AllowIrDA** are listed in the following table.

Value	Meaning
0	Disable IrDA.
1	Allow IrDA.

This element SHOULD be ignored if the client does not have the capability of transmitting or receiving infrared signals.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.10 AllowPOPIMAPEmail

The **AllowPOPIMAPEmail** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows access to POP or IMAP email.

The **AllowPOPIMAPEmail** element cannot have child elements.

Valid values for **AllowPOPIMAPEmail** are listed in the following table.

Value	Meaning
0	POP or IMAP email access is not allowed.
1	POP or IMAP email access is allowed.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes

Protocol version	Element support
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.11 AllowRemoteDesktop

The **AllowRemoteDesktop** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of Remote Desktop.

The **AllowRemoteDesktop** element cannot have child elements.

Valid values for **AllowRemoteDesktop** are listed in the following table.

Value	Meaning
0	Do not allow the use of Remote Desktop.
1	Allow the use of Remote Desktop.

This element SHOULD be ignored if the client does not support connecting remotely to a personal computer.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.12 AllowSimpleDevicePassword

The **AllowSimpleDevicePassword** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the client allows simple passwords. A simple password is one consisting only of repeated ("2222") or sequential ("abcd") characters.

The **AllowSimpleDevicePassword** element cannot have child elements.

Valid values for **AllowSimpleDevicePassword** are listed in the following table.

Value	Meaning
0	Simple passwords are not allowed.
1	Simple passwords are allowed.

If **AllowSimpleDevicePassword** is not included in a response, a client SHOULD treat this value as 1.

If the **AllowSimpleDevicePassword** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is set to FALSE (0), the client SHOULD ignore this element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.13 AllowSMIMEEncryptionAlgorithmNegotiation

The **AllowSMIMEEncryptionAlgorithmNegotiation** element is an optional child element of type **integer** ([MS-ASDTYPE] section 2.6) of the **EASProvisionDoc** element (section 2.2.2.28) that controls negotiation of the encryption algorithm.

The **AllowSMIMEEncryptionAlgorithmNegotiation** element cannot have child elements.

Valid values for **AllowSMIMEEncryptionAlgorithmNegotiation** are listed in the following table.

Value	Meaning
0	Do not negotiate.
1	Negotiate a strong algorithm.
2	Negotiate any algorithm.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.14 AllowSMIMESoftCerts

The **AllowSMIMESoftCerts** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the client can use soft certificates to sign outgoing messages.

The **AllowSMIMESoftCerts** element cannot have child elements.

Valid values for **AllowSMIMESoftCerts** are listed in the following table.

Value	Meaning
0	Soft certificates are not allowed.
1	Soft certificates are allowed.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.15 AllowStorageCard

The **AllowStorageCard** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows use of the storage card.

The **AllowStorageCard** element cannot have child elements.

Valid values for **AllowStorageCard** are listed in the following table.

Value	Meaning
0	SD card use is not allowed.
1	SD card use is allowed.

This element SHOULD be ignored if the client does not support storing data on removable storage.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.16 AllowTextMessaging

The **AllowTextMessaging** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of SMS or text messaging.

The **AllowTextMessaging** element cannot have child elements.

Valid values for **AllowTextMessaging** are listed in the following table.

Value	Meaning
0	SMS or text messaging is not allowed.
1	SMS or text messaging is allowed.

This element SHOULD be ignored if the client does not support SMS or text messaging.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.17 AllowUnsignedApplications

The **AllowUnsignedApplications** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows unsigned applications to execute.

The **AllowUnsignedApplications** element cannot have child elements.

Valid values for **AllowUnsignedApplications** are listed in the following table.

Value	Meaning
0	Unsigned applications are not allowed to execute.
1	Unsigned applications are allowed to execute.

The client SHOULD ignore the **AllowUnsignedApplications** element if the client does not execute unsigned applications.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.18 AllowUnsignedInstallationPackages

The **AllowUnsignedInstallationPackages** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows unsigned **cabinet (.cab) files** to be installed.

The **AllowUnsignedInstallationPackages** element cannot have child elements.

Valid values for **AllowUnsignedInstallationPackages** are listed in the following table.

Value	Meaning
0	Unsigned cabinet (.cab) files are not allowed to be installed.
1	Unsigned cabinet (.cab) files are allowed to be installed.

The client SHOULD ignore the **AllowUnsignedInstallationPackage** element if the client does not install applications from unsigned cabinet (.cab) files.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	

Protocol version	Element support
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.19 AllowWifi

The **AllowWifi** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device allows the use of Wi-Fi connections.

The **AllowWifi** element cannot have child elements.

Valid values for **AllowWifi** are listed in the following table.

Value	Meaning
0	The use of Wi-Fi connections is not allowed.
1	The use of Wi-Fi connections is allowed.

This element SHOULD be ignored if the client does not have Wi-Fi capability.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.20 AlphanumericDevicePasswordRequired

The **AlphanumericDevicePasswordRequired** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether a client requires an alphanumeric password.

The **AlphanumericDevicePasswordRequired** element cannot have child elements.

Valid values for **AlphanumericDevicePasswordRequired** are listed in the following table.

Value	Meaning
0	Alphanumeric device password is not required.
1	Alphanumeric device password is required.

If **AlphanumericDevicePasswordRequired** is not included in a response, a client SHOULD treat this value as 0.

If the **AlphanumericDevicePasswordRequired** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is FALSE (0), the client ignores this element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.21 ApplicationName

The **ApplicationName** element is an optional child element of type **string** ([MS-ASDTYPE] section 2.7) of the **UnapprovedInROMApplicationList** element (section 2.2.2.55) that specifies the name of an in-ROM application (.exe file) that is not approved for execution. Only in-ROM applications are valid values for this element. In-memory applications MUST be ignored.

There is no limit on the number of **ApplicationName** elements that are defined for a **UnapprovedInROMApplicationList** element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.22 ApprovedApplicationList

The **ApprovedApplicationList** element is an optional **container** ([MS-ASDTYPE] section 2.2) element that specifies a list of in-memory applications that are approved for execution. It is a child of the **EASProvisionDoc** element (section 2.2.2.28). Only in-memory applications are affected by this element. This element does not apply to in-ROM applications. If present, the client MUST only allow the in-memory applications specified by this element to execute.

A command response has a maximum of one **ApprovedApplicationList** element per **EASProvisionDoc** element.

The **ApprovedApplicationList** element has only the following child element:

- **Hash** (section 2.2.2.29): This element is optional.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.23 AttachmentsEnabled

The **AttachmentsEnabled** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether email attachments are enabled for download.

The **AttachmentsEnabled** element cannot have child elements.

Valid values for **AttachmentsEnabled** are listed in the following table.

Value	Meaning
0	Attachments are not allowed to be downloaded.
1	Attachments are allowed to be downloaded.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.24 Data

The **Data** element specifies policy settings for a device. It is either a **string** data type, as specified in [MS-ASDTYPE] section 2.7, or a **container** data type, as specified in ([MS-ASDTYPE] section 2.2, depending on the protocol version that is being used. For details, see the element definition in the following sections.

- **Data** element, **container** data type — section 2.2.2.24.1
- **Data** element, **string** data type — section 2.2.2.24.2

2.2.2.24.1 Data (container Data Type)

The **Data** element as a **container** data type ([MS-ASDTYPE] section 2.2) contains a child element in which the policy settings for a device are specified. It is a required child element of the **Policy** element (section 2.2.2.41) in responses to initial **Provision** command requests, as specified in section 3.2.5.1.1. It is not present in responses to acknowledgment requests, as specified in section 3.2.5.1.2.

This element requires that the **PolicyType** element (section 2.2.2.43) is set to "MS-EAS-Provisioning-WBXML".

As a **container** data type, the **Data** element has only the following child element:

EASProvisionDoc (section 2.2.2.28): One instance of this element is required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

The **string** data type **Data** element (section 2.2.2.24.2) is used instead of the **container** data type **Data** element with protocol version 2.5.

2.2.2.24.2 Data (string Data Type)

The **Data** element as a **string** data type ([MS-ASDTYPE] section 2.7) contains text that specifies the policy settings for a device. It is a required child element of the **Policy** element (section 2.2.2.41) in responses to initial **Provision** command requests, as specified in section 3.2.5.1.1. It is not present in responses to acknowledgment requests, as specified in section 3.2.5.1.2.

This element requires that the **PolicyType** element (section 2.2.2.43) is set to "MS-WAP-Provisioning-XML".

As a **string** data type, the value of the **Data** element is a character string that is formatted according to the WAP (Wireless Applications Protocol) Windows Mobile provisioning **XML schema**, as described in [MSDN-MSPROVDTDFormat].

The WAP Windows Mobile provisioning XML schema defines a top-level element, **wap-provisioningdoc**, and several child elements, but the string schema of the **Data** element uses only the **characteristic** element as a child element of the **wap-provisioningdoc** element. The string schema includes two top-level **characteristic** elements, which specify the "SecurityPolicy" and "Registry" configuration service providers. The nested **characteristic** elements specify branches within the "Registry" configuration service provider. Each **parm** element specifies a parameter and its

value. The following syntax block shows the string schema for the **Data** element. Details about the parameters and their values follow the syntax block.

```
<wap-provisioningdoc>
  <characteristic type="SecurityPolicy">
    <parm name="4131" value="ParmValue"/>
  </characteristic>
  <characteristic type="Registry">
    <characteristic type="HKLM\Comm\Security\Policy\LASSD\AE\{50C13377-C66D-400C-889E-
C316FC4AB374}">
      <parm name="AEFrequencyType" value="ParmValue"/>
      <parm name="AEFrequencyValue" value="ParmValue"/>
    </characteristic>
    <characteristic type="HKLM\Comm\Security\Policy\LASSD">
      <parm name="DeviceWipeThreshold" value="ParmValue"/>
      <parm name="CodewordFrequency" value="ParmValue"/>
    </characteristic>
    <characteristic type="HKLM\Comm\Security\Policy\LASSD\LAP\lap pw">
      <parm name="MinimumPasswordLength" value="ParmValue"/>
      <parm name="PasswordComplexity" value="ParmValue"/>
    </characteristic>
  </characteristic>
</wap-provisioningdoc>
```

The seven parameters and their valid values are as follows.

- *4131* — Specifies whether a password is required. The value 0 (zero) indicates that a password is required; 1 indicates that a password is not required.
- *AEFrequencyType* — Specifies whether the device will lock itself after a period of user inactivity specified by the *AEFrequencyValue* parameter. The value 0 (zero) indicates that the user determines whether to lock the device; 1 indicates that the device will lock itself.
- *AEFrequencyValue* — Specifies the number of minutes of user inactivity before the device locks. The value 0 (zero) indicates that the device locks if the screen is turned off. A value greater than 99 indicates that the user inactivity is unlimited.
- *DeviceWipeThreshold* — Specifies the maximum number of failed password logon attempts that are permitted before the device wipes itself. Once the threshold is reached, the device wipes the memory, including all data and certificates. Valid values are 4 through 16. If the *4131* parameter is set to 1, the client ignores the *DeviceWipeThreshold* parameter.
- *CodewordFrequency* — Specifies the number of times an incorrect password can be entered before a codeword is displayed. After entering the displayed codeword, the user is able to make more password attempts. The purpose of the codeword prompt is to insure that the incorrect password attempts are not the result of accidental key presses. The value is either -1, indicating that the device determines how often to prompt for the codeword, or a value that is less than the value of the *DeviceWipeThreshold* parameter.
- *MinimumPasswordLength* — Specifies the minimum length of the client password. Valid values are 1 through 18, inclusive. This value is ignored if the *4131* parameter is set to 1.
- *PasswordComplexity* — Specifies the complexity of the password. The value 0 (zero) requires the password to consist of alpha-numeric characters. The value 2 allows either numeric or alpha-numeric characters.

To insure that the contents of the **Data** element is correctly interpreted, the angle brackets "<" and ">", which are **XML** syntax markers used to enclose XML elements, MUST be represented by escape sequences: The "<" escape sequence represents the left angle bracket, and ">" the right angle bracket.

The following example shows the **Data** element with a properly formatted string.

```
<Data>&lt;wap-provisioningdoc&gt;&lt;characteristic type="SecurityPolicy"&gt;&lt;parm
name="4131" value="0"/&gt;&lt;/characteristic&gt;&lt;characteristic
type="Registry"&gt;&lt;characteristic type="HKLM\Comm\Security\Policy\LASSD\AE\{50C13377-
C66D-400C-889E-C316FC4AB374}"&gt;&lt;parm name="AEFrequencyType" value="1"/&gt;&lt;parm
name="AEFrequencyValue" value="5"/&gt;&lt;/characteristic&gt;&lt;characteristic
type="HKLM\Comm\Security\Policy\LASSD"&gt;&lt;parm name="DeviceWipeThreshold"
value="10"/&gt;&lt;parm name="CodewordFrequency"
value="3"/&gt;&lt;/characteristic&gt;&lt;characteristic
type="HKLM\Comm\Security\Policy\LASSD\LAP\lap pw"&gt;&lt;parm name="MinimumPasswordLength"
value="8"/&gt;&lt;parm name="PasswordComplexity"
value="0"/&gt;&lt;/characteristic&gt;&lt;/characteristic&gt;&lt;/wap-
provisioningdoc&gt;</Data>
```

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	
12.1	
14.0	
14.1	
16.0	
16.1	

The **container** data type **Data** element (section 2.2.2.24.1) is used instead of the **string** data type **Data** element with protocol versions 12.0, 12.1, 14.0, 14.1, 16.0 and 16.1.

2.2.2.25 DevicePasswordEnabled

The **DevicePasswordEnabled** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether a client requires a password.

The **DevicePasswordEnabled** element cannot have child elements.

Valid values for **DevicePasswordEnabled** are listed in the following table.

Value	Meaning
0	Device password is not required.
1	Device password is required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.26 DevicePasswordExpiration

The **DevicePasswordExpiration** element is an optional child element of type **unsignedIntOrEmpty** (section 2.2.3.3) of the **EASProvisionDoc** element, as specified in section 2.2.2.28, that specifies the maximum number of days until a password expires.

The **DevicePasswordExpiration** element can be empty, indicating that no password expiration policy is set.

The **DevicePasswordExpiration** element cannot have child elements.

Valid values for **DevicePasswordExpiration** are listed in the following table.

Value	Meaning
0	Passwords do not expire.
>0	Passwords expire in the specified maximum number of days.

If **DevicePasswordExpiration** is empty or is not included in a response, a client SHOULD treat this value as 0.

If the **DevicePasswordExpiration** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is set to FALSE (0), the client SHOULD ignore this element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.27 DevicePasswordHistory

The **DevicePasswordHistory** element is an optional child element of type **unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the minimum number of previously used passwords stored to prevent reuse by the client.

The **DevicePasswordHistory** element cannot have child elements.

Valid values for **DevicePasswordHistory** are listed in the following table.

Value	Meaning
0	Storage of previously used passwords is not required.
>0	The minimum number of previously used passwords to be stored.

If **DevicePasswordHistory** is not included in a response, then a client SHOULD treat this value as 0.

If the value of the **DevicePasswordHistory** element is greater than 0, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is set to TRUE (1), the client disallows the user from using a stored prior password after a password expires.

If the **DevicePasswordHistory** element is included in a response, and the value of the **DevicePasswordEnabled** element is set to FALSE (0), the client SHOULD ignore this element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes

Protocol version	Element support
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.28 EASProvisionDoc

The **EASProvisionDoc** element is a required **container** ([MS-ASDTYPE] section 2.2) element that specifies the collection of security settings for device provisioning. It is a child of the **Data** element (section 2.2.2.24.1).

A command response has a minimum of one **EASProvisionDoc** element per **Data** element.

The **EASProvisionDoc** element has only the following child elements:

- **AllowBluetooth** (section 2.2.2.2)
- **AllowBrowser** (section 2.2.2.3)
- **AllowCamera** (section 2.2.2.4)
- **AllowConsumerEmail** (section 2.2.2.5)
- **AllowDesktopSync** (section 2.2.2.6)
- **AllowHTMLEmail** (section 2.2.2.7)
- **AllowInternetSharing** (section 2.2.2.8)
- **AllowIrDA** (section 2.2.2.9)
- **AllowPOPIMAPEmail** (section 2.2.2.10)
- **AllowRemoteDesktop** (section 2.2.2.11)
- **AllowSimpleDevicePassword** (section 2.2.2.12)
- **AllowSMIMEEncryptionAlgorithmNegotiation** (section 2.2.2.13)
- **AllowSMIMESoftCerts** (section 2.2.2.14)
- **AllowStorageCard** (section 2.2.2.15)
- **AllowTextMessaging** (section 2.2.2.16)
- **AllowUnsignedApplications** (section 2.2.2.17)
- **AllowUnsignedInstallationPackages** (section 2.2.2.18)
- **AllowWifi** (section 2.2.2.19)
- **AlphanumericDevicePasswordRequired** (section 2.2.2.20)
- **ApprovedApplicationList** (section 2.2.2.22)
- **AttachmentsEnabled** (section 2.2.2.23)

- **DevicePasswordEnabled** (section 2.2.2.25)
- **DevicePasswordExpiration** (section 2.2.2.26)
- **DevicePasswordHistory** (section 2.2.2.27)
- **MaxAttachmentSize** (section 2.2.2.30)
- **MaxCalendarAgeFilter** (section 2.2.2.31)
- **MaxDevicePasswordFailedAttempts** (section 2.2.2.32)
- **MaxEmailAgeFilter** (section 2.2.2.33)
- **MaxEmailBodyTruncationSize** (section 2.2.2.34)
- **MaxEmailHTMLBodyTruncationSize** (section 2.2.2.35)
- **MaxInactivityTimeDeviceLock** (section 2.2.2.36)
- **MinDevicePasswordComplexCharacters** (section 2.2.2.37)
- **MinDevicePasswordLength** (section 2.2.2.38)
- **PasswordRecoveryEnabled** (section 2.2.2.39)
- **RequireDeviceEncryption** (section 2.2.2.46)
- **RequireEncryptedSMIMEMessages** (section 2.2.2.47)
- **RequireEncryptionSMIMEAlgorithm** (section 2.2.2.48)
- **RequireManualSyncWhenRoaming** (section 2.2.2.49)
- **RequireSignedSMIMEAlgorithm** (section 2.2.2.50)
- **RequireSignedSMIMEMessages** (section 2.2.2.51)
- **RequireStorageCardEncryption** (section 2.2.2.52)
- **UnapprovedInROMApplicationList** (section 2.2.2.55)

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes

Protocol version	Element support
16.1	Yes

2.2.2.29 Hash

The **Hash** element is an optional child element of type **string** ([MS-ASDTYPE] section 2.7) of the **ApprovedApplicationList** element (section 2.2.2.22) that specifies the SHA1 hash of an approved in-memory application. Only SHA1 hashes of in-memory applications are valid values for this element. SHA1 hashes of in-ROM applications MUST be ignored.

There is no limit on the number of **Hash** elements that are defined for a **ApprovedApplicationList** element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.30 MaxAttachmentSize

The **MaxAttachmentSize** element is an optional child element of type **unsignedIntOrEmpty** (section 2.2.3.3) of the **EASProvisionDoc** element, as specified in section 2.2.2.28, that specifies the maximum attachment size in bytes as determined by security policy.

The **EASProvisionDoc** element has at most one instance of the **MaxAttachmentSize** element. If the element is empty, the client interprets this as meaning no maximum attachment size has been set by the security policy.

The **MaxAttachmentSize** element cannot have child elements.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.31 MaxCalendarAgeFilter

The **MaxCalendarAgeFilter** element is an optional child element of type **unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the maximum number of calendar days that can be synchronized.

The **MaxCalendarAgeFilter** element cannot have child elements.

Valid values for **MaxCalendarAgeFilter** are listed in the following table.

Value	Meaning
0	All days
4	2 weeks
5	1 month
6	3 months
7	6 months

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes

Protocol version	Element support
16.0	Yes
16.1	Yes

2.2.2.32 MaxDevicePasswordFailedAttempts

The **MaxDevicePasswordFailedAttempts** element is an optional child element of type **unsignedByteOrEmpty** (section 2.2.3.2) of the **EASProvisionDoc** element, as specified in section 2.2.2.28, that specifies the maximum number of failed password logon attempts that are permitted. The client SHOULD perform a local wipe or enter a timed lock out mode if the maximum number of failed password logon attempts is reached.

The **MaxDevicePasswordFailedAttempts** element cannot have child elements.

The **MaxDevicePasswordFailedAttempts** element can be empty or have a value in the range from 4 through 16. If the element is empty or not present in a response, the client interprets this as meaning that no maximum number of failed password logon attempts has been set by the security policy.

If the **MaxDevicePasswordFailedAttempts** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is set to FALSE (0), the client ignores this element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.33 MaxEmailAgeFilter

The **MaxEmailAgeFilter** element is an optional child element of type **unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the email age limit for synchronization.

The **MaxEmailAgeFilter** element cannot have child elements.

Valid values are listed in the following table and represent the maximum allowable number of days to sync email.

Value	Meaning
0	Sync all
1	1 day
2	3 days
3	1 week
4	2 weeks
5	1 month

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.34 MaxEmailBodyTruncationSize

The **MaxEmailBodyTruncationSize** element is an optional child element of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the maximum truncation size for plain text–formatted email.

The **MaxEmailBodyTruncationSize** element cannot have child elements.

Valid values for the **MaxEmailBodyTruncationSize** element are an **integer** ([MS-ASDTYPE] section 2.6) of one of the values or ranges listed in the following table.

Value	Meaning
-1	No truncation.
0	Truncate only the header.
>0	Truncate the email body to the specified size.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.35 MaxEmailHTMLBodyTruncationSize

The **MaxEmailHTMLBodyTruncationSize** element is an optional child element of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the maximum truncation size for HTML-formatted email.

The **MaxEmailHTMLBodyTruncationSize** element cannot have child elements.

Valid values for the **MaxEmailHTMLBodyTruncationSize** element are an **integer** ([MS-ASDTYPE] section 2.6) of one of the values or ranges listed in the following table.

Value	Meaning
-1	No truncation.
0	Truncate only the header.
>0	Truncate the email body to the specified size.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	

Protocol version	Element support
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.36 MaxInactivityTimeDeviceLock

The **MaxInactivityTimeDeviceLock** element is an optional child element of type **unsignedIntOrEmpty** (section 2.2.3.3) of the **EASProvisionDoc** element, as specified in section 2.2.2.28, that specifies the maximum number of seconds of inactivity before the device locks itself.

The **MaxInactivityTimeDeviceLock** element cannot have child elements.

If this value is greater than or equal to 9999, the client interprets it as unlimited.

If the **MaxInactivityTimeDeviceLock** element is empty or not included in a response, the client interprets this as meaning that no time device lock has been set by the security policy.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.37 MinDevicePasswordComplexCharacters

The **MinDevicePasswordComplexCharacters** element is an optional child element of type **unsignedByte** ([MS-ASDTYPE] section 2.8) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the required level of complexity of the client password.

The **MinDevicePasswordComplexCharacters** element cannot have child elements.

Valid values for **MinDevicePasswordComplexCharacters** are 1 to 4. The value specifies the number of character groups that are required to be present in the password. The character groups are defined as:

- Lower case alphabetical characters
- Upper case alphabetical characters
- Numbers
- Non-alphanumeric characters

For example, if the value of **MinDevicePasswordComplexCharacters** is 2, a password with both upper case and lower case alphabetical characters would be sufficient, as would a password with lower case alphabetical characters and numbers.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.38 MinDevicePasswordLength

The **MinDevicePasswordLength** element is an optional child element of type **unsignedByteOrEmpty** (section 2.2.3.2) of the **EASProvisionDoc** element, as specified in section 2.2.2.28, that specifies the minimum client password length.

The **MinDevicePasswordLength** element cannot have child elements.

The **MinDevicePasswordLength** element can be empty or have a value no less than 1 and no greater than 16. If the element is empty or the value of this element is 1, there is no minimum length for the device password.

If the **MinDevicePasswordLength** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is FALSE (0), the client SHOULD ignore this element.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-

ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.39 PasswordRecoveryEnabled

The **PasswordRecoveryEnabled** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the server supports storage of a recovery password to be sent by the client using the **Settings** command.

The **PasswordRecoveryEnabled** element cannot have child elements.

Valid values for **PasswordRecoveryEnabled** are listed in the following table.

Value	Meaning
0	Password recovery is not enabled on the server.
1	Password recovery is enabled on the server.

A recovery password is a special password created by the client that gives the administrator or user the ability to log on to the device one time, after which the user is required to create a new password. The client then creates a new recovery password. If the **PasswordRecoveryEnabled** element is set to 1 (TRUE), the server supports storage of a recovery password sent by the device. If the element is set to 0 (FALSE), the device SHOULD NOT send a recovery password, because the server does not support storage of the password.

If **PasswordRecoveryEnabled** is not included in a response, a client SHOULD treat this value as 0.

If the **PasswordRecoveryEnabled** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.25) is FALSE (0), the client SHOULD ignore this element. This element SHOULD be ignored if the client does not support recovery passwords.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.40 Policies

The **Policies** element is a required **container** ([MS-ASDTYPE] section 2.2) element that specifies a collection of security policies. It is a child of the **Provision** element (section 2.2.2.44).

The **Policies** element has only the following child element:

- **Policy** (section 2.2.2.41): At least one element of this type is required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.41 Policy

The **Policy** element is a required **container** ([MS-ASDTYPE] section 2.2) element that specifies a policy. It is a child of the **Policies** element (section 2.2.2.40).

This element is valid in both a command request and a command response.

In the initial Provision command request, the **Policy** element has the following child element:

- **PolicyType** (section 2.2.2.43), required

In the initial Provision command response, the **Policy** element has the following child elements:

- **PolicyType** (section 2.2.2.43), required
- **PolicyKey** (section 2.2.2.42), required
- **Status** (section 2.2.2.54.1), required
- **Data** (section 2.2.2.24), required

In the acknowledgment Provision command request, the **Policy** element has the following child elements:

- **PolicyType** (section 2.2.2.43), required
- **PolicyKey** (section 2.2.2.42), required and MUST appear before the **Status** element
- **Status** (section 2.2.2.54.1), required

In the acknowledgment Provision command response, the **Policy** element has the following child elements:

- **PolicyType** (section 2.2.2.43), required
- **PolicyKey** (section 2.2.2.42), required
- **Status** (section 2.2.2.54.1), required

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.42 PolicyKey

The **PolicyKey** element is an optional element of type **string** ([MS-ASDTYPE] section 2.7) with a maximum of 64 characters and no child elements. It is a child element of the **Policy** element (section 2.2.2.41).

The value of the **PolicyKey** element SHOULD be a string representation of a 32-bit unsigned integer. **PolicyKey** is used by the server to mark the state of policy settings on the client in the settings download phase of the **Provision** command. When the client issues an initial **Provision** command, the **PolicyKey** tag and X-MS-PolicyKey are not included in the **HTTP** header. In the acknowledgement phase, the **PolicyKey** element is used by the client and server to correlate acknowledgements to a particular policy setting.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.43 PolicyType

The **PolicyType** element is a child element of type **string** ([MS-ASDTYPE] section 2.7) of the **Policy** element (section 2.2.2.41) that, in the download policy settings phase, specifies the format in which the policy settings are to be provided to the client device.

The value of the **PolicyType** element MUST be one of the values specified in the following table.

Value	Meaning
MS-WAP-Provisioning-XML	The contents of the **Data** element are formatted according to the WAP Windows Mobile provisioning **XML** schema, as specified in section 2.2.2.24.2.
MS-EAS-Provisioning-WBXML	The contents of the **Data** element are formatted according to the Exchange ActiveSync provisioning **WBXML** schema, as specified in section 2.2.2.24.1.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

The value "MS-WAP-Provisioning-XML" is used with protocol version 2.5. The value "MS-EAS-Provisioning-WBXML" is used with protocol versions 12.0, 12.1, 14.0, 14.1, 16.0 and 16.1.

2.2.2.44 Provision

The **Provision** element is a required **container** ([MS-ASDTYPE] section 2.2) element in a provisioning request and response that specifies the capabilities and permissions of a device.

The **Provision** element has the following child elements:

- **settings:DeviceInformation** (section 2.2.2.53)
- **Status** (section 2.2.2.54.2)
- **Policies** (section 2.2.2.40)
- **RemoteWipe** (section 2.2.2.45)
- **AccountOnlyRemoteWipe** (section 2.2.2.1)

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.45 RemoteWipe

The **RemoteWipe** element is an optional **container** ([MS-ASDTYPE] section 2.2) element that specifies either a remote wipe directive from the server or a client's confirmation of a server's remote wipe directive.

A server response MUST NOT include any child elements in the **RemoteWipe** element.

The **RemoteWipe** element is sent in a command request only in response to a remote wipe directive from the server.

The **RemoteWipe** element has the following child element in a command request:

- **Status** (section 2.2.2.54.3): One element of this type is required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.46 RequireDeviceEncryption

The **RequireDeviceEncryption** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the client uses encryption.

The **RequireDeviceEncryption** element cannot have child elements.

Valid values for **RequireDeviceEncryption** are listed in the following table.

Value	Meaning
0	Encryption is not required.
1	Encryption is required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.47 RequireEncryptedSMIMEMessages

The **RequireEncryptedSMIMEMessages** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the client sends encrypted email messages.

The **RequireEncryptedSMIMEMessages** element cannot have child elements.

Valid values for **RequireEncryptedSMIMEMessages** are listed in the following table.

Value	Meaning
0	Encrypted email messages are not required.
1	Email messages are required to be encrypted.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes

Protocol version	Element support
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.48 RequireEncryptionSMIMEAlgorithm

The **RequireEncryptionSMIMEAlgorithm** element is an optional child element of type **integer** ([MS-ASDTYPE] section 2.6) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the algorithm used when encrypting S/MIME messages.

The **RequireEncryptionSMIMEAlgorithm** element cannot have child elements.

Valid values for **RequireEncryptionSMIMEAlgorithm** are listed in the following table.

Value	Meaning
0	TripleDES algorithm
1	DES algorithm
2	RC2128bit
3	RC264bit
4	RC240bit

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.49 RequireManualSyncWhenRoaming

The **RequireManualSyncWhenRoaming** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device requires manual synchronization when the device is roaming.

The **RequireManualSyncWhenRoaming** element cannot have child elements.

Valid values for **RequireManualSyncWhenRoaming** are listed in the following table.

Value	Meaning
0	Do not require manual sync; allow direct push when roaming.
1	Require manual sync when roaming.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.50 RequireSignedSMIMEAlgorithm

The **RequireSignedSMIMEAlgorithm** element is an optional child element of type **integer** ([MS-ASDTYPE] section 2.6) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies the algorithm used when signing S/MIME messages.

The **RequireSignedSMIMEAlgorithm** element cannot have child elements.

Valid values for **RequireSignedSMIMEAlgorithm** are listed in the following table.

Value	Meaning
0	Use SHA1.
1	Use MD5.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.51 RequireSignedSMIMEMessages

The **RequireSignedSMIMEMessages** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the client sends signed S/MIME messages.

The **RequireSignedSMIMEMessages** element cannot have child elements.

Valid values for **RequireSignedSMIMEMessages** are listed in the following table.

Value	Meaning
0	Signed S/MIME messages are not required.
1	Signed S/MIME messages are required.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes

Protocol version	Element support
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.52 RequireStorageCardEncryption

The **RequireStorageCardEncryption** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.28) that specifies whether the device encrypts content that is stored on the storage card.

The **RequireStorageCardEncryption** element cannot have child elements.

Valid values for **RequireStorageCardEncryption** are listed in the following table.

Value	Meaning
0	Encryption of the device storage card is not required.
1	Encryption of the device storage card is required.

This element SHOULD be ignored if the client does not support storing data on removable storage.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.53 settings:DeviceInformation

The **settings:DeviceInformation** element is an optional **container** ([MS-ASDTYPE] section 2.2) element that is used for sending the client device's properties to the server in an initial **Provision** command request. It is a child of the **Provision** element (section 2.2.2.44). The

settings:DeviceInformation element is defined in the **Settings** XML namespace, as specified in [MS-ASCMD] section 2.2.3.45.

When the **Provision** command is used to send the **settings:DeviceInformation** element, it sends the information about the client device to the server, as specified for the **settings:DeviceInformation** element under the **Settings** command in [MS-ASCMD] section 2.2.1.18.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	
14.0	
14.1	Yes
16.0	Yes
16.1	Yes

When protocol version 14.1, 16.0, or 16.1 is used, the client MUST send the **settings:DeviceInformation** element with its contents when sending an initial **Provision** command request to the server but not on subsequent requests. The **settings:DeviceInformation** element MUST contain a **settings:Set** child element ([MS-ASCMD] section 2.2.3.167), and the **settings:Set** element MUST at least contain a **settings:Model** child element ([MS-ASCMD] section 2.2.3.115).

When protocol version 14.0, 12.1, or 12.0 is used, the client MUST NOT send the **settings:DeviceInformation** element in any **Provision** command request. In these cases, the **settings:DeviceInformation** element can be used in a **Settings** command request, as specified in [MS-ASCMD] section 3.1.5.2.

Protocol version 2.5 does not support sending device information to the server.

2.2.2.54 Status

The **Status** element is a child element of the **Policy** element (section 2.2.2.41), the **Provision** element (section 2.2.2.44), and the **RemoteWipe** element (section 2.2.2.45). The definition of this element differs according to the context in which it is used. For more details, see section 2.2.2.54.1, section 2.2.2.54.2, and section 2.2.2.54.3.

2.2.2.54.1 Status (Policy)

The **Status** element is a required child of the **Policy** element in command responses and an optional child of the **Policy** element in command requests.

In a command response, the value of this element is an **unsignedByte** ([MS-ASDTYPE] section 2.8). The value indicates the success or failure of a client's initial request to retrieve policy settings from the

server. The following table lists valid values for the **Status** element when it is the child of the **Policy** element in the response from the server to the client.

Value	Meaning
1	Success.
2	There is no policy for this client.
3	Unknown **PolicyType** value.
4	The policy data on the server is corrupted (possibly tampered with).
5	The client is acknowledging the wrong **policy key**.

In a command request, the value of this element is a **string** ([MS-ASDTYPE] section 2.7). The value indicates the success or failure of the client to apply the policy settings retrieved from the server. The following table lists valid values for the **Status** element when it is the child of the **Policy** element in the request from the client to the server.

Value	Meaning
1	Success
2	Partial success (at least the PIN was enabled).
3	The client did not apply the policy at all.
4	The client claims to have been provisioned by a third party.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.54.2 Status (Provision)

The **Status** element is a required child element of the **Provision** element in command responses. The value of this element is an **unsignedByte** ([MS-ASDTYPE] section 2.8). The value indicates the success or failure of the **Provision** command.

The following table lists values for the **Status** element when it is the child of the **Provision** element. For details about status values common to all ActiveSync commands, see [MS-ASCMD] section 2.2.2.

Value	Meaning
1	Success
2	Protocol error
3	General server error

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.54.3 Status (RemoteWipe)

The **Status** element is a required child of the **RemoteWipe** or **AccountOnlyRemoteWipe** element in command requests. The value of this element is an **unsignedByte** ([MS-ASDTYPE] section 2.8). The value indicates the success or failure of a remote wipe operation on the client. The following table lists valid values for the **Status** element when it is the child of the **RemoteWipe** or **AccountOnlyRemoteWipe** element.

Value	Meaning
1	The client remote wipe operation was successful.
2	The remote wipe operation failed.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	Yes
12.0	Yes
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.2.55 UnapprovedInROMApplicationList

The **UnapprovedInROMApplicationList** element is an optional **container** ([MS-ASDTYPE] section 2.2) element that specifies a list of in-ROM applications that are not approved for execution. It is a child of the **EASProvisionDoc** element (section 2.2.2.28). Only applications that are preinstalled in ROM are affected by the entries in this element. This element does not apply to applications that are installed in-memory.

A command response has a maximum of one **UnapprovedInROMApplicationList** element per **EASProvisionDoc** element.

The **UnapprovedInROMApplicationList** element has only the following child element:

- **ApplicationName** (section 2.2.2.21): This element is optional.

Protocol Versions

The following table specifies the protocol versions that support this element. The client indicates the protocol version being used by setting either the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.6, or the **Protocol version** field, as specified in [MS-ASHTTP] section 2.2.1.1.1.1, in the request.

Protocol version	Element support
2.5	
12.0	
12.1	Yes
14.0	Yes
14.1	Yes
16.0	Yes
16.1	Yes

2.2.3 Simple Types

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

Simple type	Description
EmptyVal (section 2.2.3.1)	A type that cannot contain a value.
unsignedByteOrEmpty (section 2.2.3.2)	A type that can either be an **xs:unsignedByte** ([XMLSCHEMA2/2] section 3.3.24) or empty.
unsignedIntOrEmpty (section 2.2.3.3)	A type that can either be an **xs:unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) or empty.

2.2.3.1 EmptyVal Simple Type

The **EmptyVal** simple type represents an empty value.

```
<xs:simpleType name="EmptyVal">
  <xs:restriction base="xs:string">
    <xs:maxLength value="0"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.3.2 unsignedByteOrEmpty Simple Type

The **unsignedByteOrEmpty** simple type represents a value that can either be an **xs:unsignedByte** type, as specified in [XMLSCHEMA2/2] section 3.3.24, or an empty value.

```
<xs:simpleType name="unsignedByteOrEmpty">
  <xs:union memberTypes="xs:unsignedByte EmptyVal"/>
</xs:simpleType>
```

2.2.3.3 unsignedIntOrEmpty Simple Type

The **unsignedIntOrEmpty** simple type represents a value that can either be an **xs:unsignedInt** type, as specified in [XMLSCHEMA2/2] section 3.3.22, or an empty value.

```
<xs:simpleType name="unsignedIntOrEmpty">
  <xs:union memberTypes="xs:unsignedInt EmptyVal"/>
</xs:simpleType>
```

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The following figure shows the process for downloading policy settings.

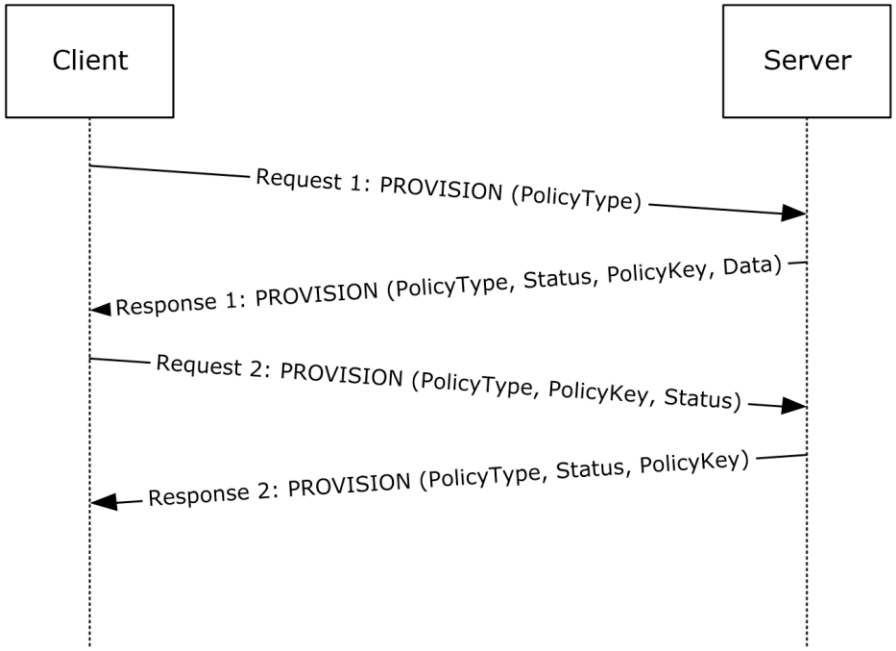

Figure 1: Downloading policy settings

The following table lists the command sequence for downloading policy settings.

Order	Client action	Server action
1	The client sends a **Provision** command request with the type of policy settings to be downloaded.	The server response contains the policy type, policy key, data, and status code.
2	The client acknowledges that it received and applied the policy settings by sending another **Provision** command request with the policy type, policy key, and status code.	The server response contains the policy type, policy key, and status code to indicate that the server recorded the client's acknowledgement.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Provision Command

The **Provision** command enables client devices to send the server information about the device, to request from the server the security policy settings set by the server administrator, and to report on the status of a remote wipe or an account only remote wipe directive.

The provisioning process has two phases: an initial phase consisting of a **Provision** command request sent by the client followed by an initial server response, then an acknowledgment phase consisting of a **Provision** command request sent by the client with an acknowledgment of the initial server response, followed by another server response.

Clients SHOULD<1> begin the provisioning process in the following situations:

- When contacting the server for the first time.
- When the server's response to any command indicates that the client needs to re-provision. Whether the server specifies this condition by returning a value in the **Status** element or by returning an **HTTP** 4xx or 5xx response code depends on the protocol version. For details, see the table of status values in [MS-ASCMD] section 2.2.2.
- When the server's response to any command indicates that the client needs to do a remote wipe. Whether the server specifies this condition by returning a value in the **Status** element or by returning an HTTP 4xx or 5xx response code depends on the protocol version. For details, see the table of status values in [MS-ASCMD] section 2.2.2.

The format of the **Provision** command request and response differs based on the context in which it is used. The contexts for the **Provision** command are:

- The initial request, as specified in section 3.1.5.1.1.
- Acknowledging security policy settings, as specified in section 3.1.5.1.2.1.
- Acknowledging a remote wipe directive, as specified in section 3.1.5.1.2.2, or an account only remote wipe directive, as specified in section 3.1.5.1.2.3.

The current security policy settings on the client are represented by the current policy key, which is sent to the server in the **X-MS-PolicyKey** header ([MS-ASHTTP] section 2.2.1.1.2.8) if the client is using a plain text query value, as specified in [MS-ASHTTP] section 2.2.1.1.1.2, or the **Policy key** field of the base64 encoded query value ([MS-ASHTTP] section 2.2.1.1.1.1) if the client is using a base64 encoded query value. The policy key is sent to the server for all protocol command requests except the **Autodiscover** command ([MS-ASCMD] section 2.2.1.1), the **Ping** command ([MS-ASCMD] section 2.2.1.13), and the **HTTP OPTIONS** command ([MS-ASHTTP] section 2.2.3).

3.1.5.1.1 Initial Request

The client sends an initial provisioning request either to retrieve the current security policy settings or in response to the server's remote wipe or account only remote wipe directive. During the initial provisioning request, the current policy key MUST be reset to 0 (zero).

To request the current security policy settings from the server, the client sends the initial provisioning request in the following format. The inclusion of the **settings:DeviceInformation** element depends on the protocol version that is being used. For details, see section 2.2.2.53.

```
<Provision>
   <settings:DeviceInformation>
      ...
   </settings:DeviceInformation>
   <Policies>
      <Policy>
         <PolicyType>...</PolicyType>
      <Policy>
   </Policies>
</Provision>
```

If the initial provisioning request is in response to receiving a status code from the server indicating that a remote wipe is requested, the initial provisioning request SHOULD consist of an empty **Provision** element (section 2.2.2.44). If the server response contains a **RemoteWipe** (section 2.2.2.45) or an **AccountOnlyRemoteWipe** (section 2.2.2.1) element within the **Provision** element, the client SHOULD acknowledge the remote wipe, as specified in section 3.1.5.1.2.2, or account only remote wipe, as specified in section 3.1.5.1.2.3. For a remote wipe, the client SHOULD then destroy all data on the device and restore it to factory default settings. For an account only remote wipe, the client SHOULD then destroy all data that it has ever received from the server and erase any stored credentials used to access the server.

If the server response includes a **Status** element (section 2.2.2.54.2) within the **Provision** element that indicates success, and also contains a **Policies** element (section 2.2.2.40) within the **Provision** element, the client ensures that the security policy settings contained in the **Policy** element (section 2.2.2.41) are actually enforced, and acknowledges the security policy settings, as specified in section 3.1.5.1.2.1. Any elements that the client ignores because the client does not support the associated feature SHOULD be considered enforced. The value of the **PolicyKey** element (section 2.2.2.42) contained within this **Policy** element is a temporary policy key that is only valid for the acknowledgment request.

The client SHOULD ignore any **Policy** element that has its **PolicyType** child element (section 2.2.2.43) set to a value that is not supported by the protocol version that is specified in the MS-ASProtocolVersion header. For details about the MS-ASProtocolVersion header, see [MS-ASHTTP] section 2.2.1.1.2.6.

3.1.5.1.1.1 Enforcing Password Requirements

The following elements represent the password requirements specified by a security policy:

- **AllowSimpleDevicePassword** (section 2.2.2.12)
- **AlphanumericDevicePasswordRequired** (section 2.2.2.20)
- **DevicePasswordEnabled** (section 2.2.2.25)
- **DevicePasswordExpiration** (section 2.2.2.26)
- **DevicePasswordHistory** (section 2.2.2.27)
- **MaxDevicePasswordFailedAttempts** (section 2.2.2.32)
- **MinDevicePasswordComplexCharacters** (section 2.2.2.37)
- **MinDevicePasswordLength** (section 2.2.2.38)
- **PasswordRecoveryEnabled** (section 2.2.2.39)

The client uses the following rules to enforce password requirements.

1. If the **DevicePasswordEnabled** element is missing or set to 0, the client SHOULD ignore the other password requirement elements.

2. The client SHOULD configure the device on which the client application is installed to require a password that meets all of the password requirements. If it does not configure the device to require the password, it MUST instead require a password that meets the requirements to access the client application and any data that the client has received from the server.

3.1.5.1.1.2 Enforcing RequireDeviceEncryption

If the **RequireDeviceEncryption** element (as specified in section 2.2.2.46) is present and set to 1, the client SHOULD configure the device on which the client application is installed to encrypt all local storage. If it does not configure the device to encrypt all local storage, it MUST encrypt all data that the client has received from the server.

3.1.5.1.2 Acknowledgment Request

The second phase of the provisioning process, the acknowledgment phase, is either an acknowledgment of security policy settings (section 3.1.5.1.2.1), or an acknowledgment of a remote wipe directive (section 3.1.5.1.2.2).

3.1.5.1.2.1 Acknowledging Security Policy Settings

During the security policy settings acknowledgment request, the current policy key MUST be set to the temporary policy key obtained from the server response to the initial request, as specified in section 3.1.5.1.1.

Clients include a security policy settings acknowledgment in the **Provision** command request sent immediately following the server response to a server policy settings request. A security policy settings acknowledgment uses the following format.

```
<Provision>
   <Policies>
      <Policy>
         <PolicyKey>...</PolicyKey>
         <Status>...</Status>
         <PolicyType>...</PolicyType>
      <Policy>
   </Policies>
</Provision>
```

The value of the **PolicyKey** element (section 2.2.2.42) MUST be set to the temporary policy key obtained from the server response to the initial request.

The client sets the value of the **Status** element to indicate the result of enforcement of the security policy, as specified in section 2.2.2.54.1.

If the server response includes a **Status** element (section 2.2.2.54.2) within the **Provision** element that indicates success, and also contains a **Policies** element (section 2.2.2.40) within the **Provision** element, the client checks for a **Policy** element (section 2.2.2.41) that has a **PolicyType** child element (section 2.2.2.43). Any **Policy** element that has a **PolicyType** child element with a value other than those specified in section 2.2.2.43 SHOULD be ignored.

The value of the **PolicyKey** element contained within this **Policy** element is a permanent policy key that is valid for subsequent command requests.

3.1.5.1.2.2 Acknowledging a Remote Wipe Directive

Clients include a remote wipe acknowledgment in the **Provision** command request sent immediately following a **Provision** command response that includes a **RemoteWipe** element (section 2.2.2.45) within the **Provision** element in the XML body. A remote wipe acknowledgment uses the following format.

```
<Provision>
   <RemoteWipe>
      <Status>...</Status>
   </RemoteWipe>
</Provision>
```

The client sets the value of the **Status** element (section 2.2.2.54.3) to indicate the result of the remote wipe. The client SHOULD then destroy all data contained on the device, returning it to original factory settings. If it does not destroy all data contained on the device, the client MUST destroy all data that it has ever received from the server and erase any stored credentials used to access the server. The client SHOULD NOT wait for or rely on any specific response from the server before proceeding with the remote wipe.

3.1.5.1.2.3 Acknowledging an Account Only Remote Wipe Directive

Clients include an account only remote wipe acknowledgment in the **Provision** command request sent immediately following a **Provision** command response that includes an **AccountOnlyRemoteWipe** element (section 2.2.2.1) within the **Provision** element in the XML body. An account only remote wipe acknowledgment uses the following format.

```
<Provision>
   <AccountOnlyRemoteWipe>
      <Status>...</Status>
   </AccountOnlyRemoteWipe>
</Provision>
```

The client sets the value of the **Status** element (section 2.2.2.54.3) to acknowledge the account only remote wipe directive. The client SHOULD then destroy all data that it has ever received from the server and erase any stored credentials used to access the server. The client SHOULD NOT wait for or rely on any specific response from the server before proceeding with the remote wipe.

3.1.5.2 Provision Command Errors

The following table specifies the actions a client SHOULD take based upon the value of the **Status** element that is a child of the **Provision** element. Status codes greater than 100 are not supported by all protocol versions. For more details, see [MS-ASCMD] section 2.2.2.

Code	Meaning	Cause	Resolution
1	Success.	The **Policies** element contains information about security policies.	Apply the applicable policy.
2	Protocol error.	Syntax error in the Provision command request.	Fix syntax in the request and resubmit.
3	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Retry.
139	The client cannot fully comply with all requirements of the policy.	The client returned a value of 2 in the **Status** child element of the **Policy** element in a request to the server to acknowledge a policy.<2> The server is	Server administrator intervention is required.

Code	Meaning	Cause	Resolution
		configured to not allow clients that cannot fully apply the policy.	
141	The device is not provisionable.	The client did not submit a policy key value in a request. The server is configured to not allow clients that do not submit a policy key value.	Include a policy key value in the X-MS-PolicyKey header ([MS-ASHTTP] section 2.2.1.1.2.8) or the **Policy key** field of the Base64 Encoded Query Value ([MS-ASHTTP] section 2.2.1.1.1.1).
145	The client is externally managed.	The client returned a value of 4 in the **Status** child element of the **Policy** element in a request to the server to acknowledge a policy. The server is configured to not allow externally managed clients.	The client can issue a new **Provision** request and apply the policy, overwriting any external provisioning. If this is not possible, server administrator intervention is required.

The following table specifies the actions a client SHOULD take based upon the value of the **Status** element that is a child of the **Policy** element. For details about the **Status** element as a child of the **Policy** element, see section 2.2.2.54.1.

Code	Meaning	Cause	Resolution
1	Success.	The requested policy data is included in the response.	Apply the policy.
2	Policy not defined.	No policy of the requested type is defined on the server.	Stop sending policy information. No policy is implemented.
3	The policy type is unknown.	The client sent a policy that the server does not recognize.	Issue a request with the **PolicyType** element set as specified in section 2.2.2.43.
4	Policy data is corrupt.	The policy data on the server is corrupt.	Server administrator intervention is required.
5	Policy key mismatch.	The client is trying to acknowledge an out-of-date or invalid policy.	Issue a new **Provision** request to obtain a valid policy key.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

See section 3.1.1 for more details.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Provision Command

The **Provision** command enables servers to obtain device information from client devices, to send security policy settings set by the server administrator and set a shared policy key, and to send a remote wipe or an account only remote wipe directive.

The server SHOULD require that the client device has requested and acknowledged the security policy settings before the client is allowed to synchronize with the server, unless a security policy is set on the server to allow it. The server relies on the client to apply the security policy settings on the client device.

The **Provision** command has two phases: an initial phase consisting of a client request followed by an initial server response, then an acknowledgment phase consisting of a client request with an acknowledgment of the initial server response, followed by another server response.

The format of the **Provision** command request and response differs based on the context in which it is used. The contexts for the **Provision** command are:

- The initial request, as specified in section 3.2.5.1.1.
- Acknowledging security policy settings, as specified in section 3.2.5.1.2.1.
- Acknowledging a remote wipe directive, as specified in section 3.2.5.1.2.2, or account only remote wipe directive, as specified in section 3.2.5.1.2.3.

The server generates, stores, and sends the policy key when it responds to a **Provision** command request for security policy settings. The current policy key on the client represents the current security policy settings.

The current policy key is received from the client for all protocol command requests except the **Autodiscover** command ([MS-ASCMD] section 2.2.1.1), the **Ping** command ([MS-ASCMD] section 2.2.1.13), and the **HTTP OPTIONS** command ([MS-ASHTTP] section 2.2.3). The current policy key SHOULD be received from the client in the **X-MS-PolicyKey** header ([MS-ASHTTP] section 2.2.1.1.2.8) if the client is using a plain text query value, as specified in [MS-ASHTTP] section 2.2.1.1.1.2, or the **Policy key** field of the base64 encoded query value ([MS-ASHTTP] section 2.2.1.1.1.1) if the client is using a base64 encoded query value.

If the policy key received from the client does not match the stored policy key on the server, or if the server determines that policy settings need to be updated on the client, the server SHOULD return a status code, as specified in [MS-ASCMD] section 2.2.2, in the next command response indicating that

the client needs to send another **Provision** command to request the security policy settings and obtain a new policy key.

3.2.5.1.1 Responding to an Initial Request

The server SHOULD store the device information that was specified in the **settings:DeviceInformation** element (section 2.2.2.53) sent by the client device and SHOULD respond to an initial security policy settings **Provision** command request with a response in the following format. The contents of the **PolicyType** element (section 2.2.2.43) and the **Data** element (section 2.2.2.24) depend on the protocol version that is being used. The **settings:DeviceInformation** element is not supported by some protocol versions. For details about these elements and protocol versions, see the element definitions.

```
<Provision>
   <settings:DeviceInformation>
      <settings:Status>...</settings:Status>
   </settings:DeviceInformation>
   <Status>...</Status>
   <Policies>
      <Policy>
         <PolicyType>...</PolicyType>
         <Status>...</Status>
         <PolicyKey>...</PolicyKey>
         <Data>
               ...
         </Data>
      </Policy>
   </Policies>
</Provision>
```

The value of the **PolicyKey** element (section 2.2.2.42) is a temporary policy key that will be valid only for an acknowledgment request to acknowledge the policy settings contained in the **Data** element.

When a policy setting that was previously set is unset on the server, the server SHOULD specify the element that represents the setting as an empty tag or a default value. In these cases, the client SHOULD either unset these values if they were previously set, or leave the setting unchanged.

The server SHOULD respond to an empty initial **Provision** command request with a response in the following format. The **RemoteWipe** or **AccountOnlyRemoteWipe** MUST only be included if a remote wipe or an account only remote wipe has been requested for the client; otherwise, it MUST be omitted.

```
<Provision>
   <Status>...</Status>
   <RemoteWipe/>
</Provision>


or

<Provision>
    <Status>…</Status>
    <AccountOnlyRemoteWipe/>
</Provision>

or

<Provision>
    <Status>…</Status>
</Provision>
```

3.2.5.1.2 Responding to an Acknowledgment Request

The second phase of the provisioning process, the acknowledgment phase, is either an acknowledgment of security policy settings (section 3.2.5.1.2.1), or an acknowledgment of a remote wipe directive (section 3.2.5.1.2.2).

3.2.5.1.2.1 Responding to a Security Policy Settings Acknowledgment

The server MUST ensure that the current policy key sent by the client in a security policy settings acknowledgment matches the temporary policy key issued by the server in the response to the initial request from this client. If it does not, the server SHOULD return a **Status** (section 2.2.2.54.2) value of 5, as specified in section 3.2.5.2.

If the policy key matches the temporary policy key, the server SHOULD check the value of the **Status** element (section 2.2.2.54.1) sent by the client in the acknowledgment to determine the client's reported level of compliance with the security policy. If the level of compliance does not meet the server's requirements, the server SHOULD return an appropriate value in the **Status** (section 2.2.2.54.2) element.

If the level of compliance meets the server's requirements, the server response is in the following format.

```
<Provision>
   <Status>...</Status>
   <Policies>
      <Policy>
         <PolicyType>...</PolicyType>
         <Status>...</Status>
         <PolicyKey>...</PolicyKey>
      </Policy>
   </Policies>
</Provision>
```

The value of the **PolicyKey** element (section 2.2.2.42) is a permanent policy key that is valid for subsequent command requests from the client.

3.2.5.1.2.2 Responding to a Remote Wipe Directive Acknowledgment

The server SHOULD record the status of the remote wipe reported by the client in the **Status** element (section 2.2.2.54.3) of the acknowledgment request. If the client reports success, the server SHOULD return a value of 1 in the **Status** element (section 2.2.2.54.2).<3> If the client reports failure, the server SHOULD return a value of 2 in the **Status** element and a remote wipe directive.<4>

The server's response is in the following format.

```
<Provision>
   <Status>...</Status>
   <RemoteWipe/>
</Provision>
```

3.2.5.1.2.3 Responding to an Account Only Remote Wipe Directive Acknowledgement

The server SHOULD record the status of the account only remote wipe reported by the client in the **Status** element (section 2.2.2.54.3) of the acknowledgment request. If the client reports success, the server SHOULD return a value of 1 in the **Status** element (section 2.2.2.54.2). If the client reports failure, the server SHOULD return a value of 2 in the **Status** element and an account only remote wipe directive.

The server's response is in the following format.

```
<Provision>
   <Status>...</Status>
   <AccountOnlyRemoteWipe/>
</Provision>
```

3.2.5.2 Provision Command Errors

Code	Meaning	Cause	Scope	Resolution
1	Success.	The requested policy data is included in the response.	Policy	Apply the policy.
2	Protocol error.	Syntax error in the **Provision** command request.	Global	Fix bug in client code.
2	Policy not defined.	No policy of the requested type is defined on the server.	Policy	Stop sending policy information. No policy is implemented.
3	The policy type is unknown.	The client sent a policy that the server does not recognize.	Policy	Issue a request with the **PolicyType** element set as specified in section 2.2.2.43.
3	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry.
5	Policy key mismatch.	The client is trying to acknowledge an out-of-date or invalid policy.	Policy	Issue a new **Provision** request to obtain a valid policy key.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

For the sake of clarity, the example request/responses do not show the **base64 encoding** of the **URI** query parameters and WBXML-encoding of the XML bodies.

4.1 Downloading the Current Server Security Policy

This section provides a walk-through of the messages that are used to download the current server security policy. This section contains the following:

- Phase 1: Enforcement
- Phase 2: Client Downloads Policy from Server
- Phase 3: Client Acknowledges Receipt and Application of Policy Settings
- Phase 4: Client Performs **FolderSync** by Using the Final **PolicyKey**

4.1.1 Phase 1: Enforcement

In the following example, the client tries the **FolderSync** command, which is denied by the server because the server has determined that the client does not have the current policy (as denoted by the X-MS-PolicyKey header). The server returns HTTP 200 (ok) with a global status code in the body of the response of 142.

Request

```
POST /Microsoft-Server-
ActiveSync?User=deviceuser&DeviceId=6F24CAD599A5BF1A690246B8C68FAE8D&DeviceType=PocketPC&Cmd=
FolderSync HTTP/1.1
Accept-Language: en-us
MS-ASProtocolVersion: 14.0
Content-Type: application/vnd.ms-sync.wbxml
X-MS-PolicyKey: 0
User-Agent: ASOM
Host: EXCH-B-003
<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
  <SyncKey>0</SyncKey>
</FolderSync>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Mon, 01 May 2006 20:15:15 GMT
Content-Length: 15

<?xml version="1.0" encoding="utf-8"?>
<FolderSync
            xmlns="FolderHierarchy:">
  <Status>142</Status>
</FolderSync>
```

4.1.2 Phase 2: Client Downloads Policy from Server

In this phase, the client downloads the policy from the server and receives a temporary policy key through the **PolicyKey** element (section 2.2.2.42). The client then uses the policy key to acknowledge

the policy and obtain a key that enables the client to successfully execute protocol commands against the server. On this initial request, the client also supplies a **settings:DeviceInformation** element (section 2.2.2.53) that describes the device.

Request

```
POST /Microsoft-Server-
ActiveSync?User=deviceuser&DeviceId=6F24CAD599A5BF1A690246B8C68FAE8D&DeviceType=PocketPC&Cmd=
Provision HTTP/1.1
Accept-Language: en-us
MS-ASProtocolVersion: 14.0
Content-Type: application/vnd.ms-sync.wbxml
X-MS-PolicyKey: 0
User-Agent: ASOM
Host: EXCH-B-003

<?xml version="1.0" encoding="utf-8"?>
<Provision xmlns="Provision:" xmlns:settings="Settings:">
    <settings:DeviceInformation>
        <settings:Set>
            <settings:Model>...</settings:Model>
            <settings:IMEI>...</settings:IMEI>
            <settings:FriendlyName>...</settings:FriendlyName>
            <settings:OS>...</settings:OS>
            <settings:OSLanguage>...</settings:OSLanguage>
            <settings:PhoneNumber>...</settings:PhoneNumber>
            <settings:MobileOperator>...</settings:MobileOperator>
            <settings:UserAgent>...</settings:UserAgent>
        </settings:Set>
    </settings:DeviceInformation>
     <Policies>
          <Policy>
               <PolicyType>MS-EAS-Provisioning-WBXML</PolicyType>
          </Policy>
     </Policies>
</Provision>
```

Response

```
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 1069
Date: Mon, 01 May 2006 20:15:15 GMT
Content-Type: application/vnd.ms-sync.wbxml
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
MS-Server-ActiveSync: 8.0
Cache-Control: private

<?xml version="1.0" encoding="utf-8"?>
<Provision xmlns="Provision:" xmlns:settings="Settings:">
     <settings:DeviceInformation>
          <settings:Status>1</settings:Status>
     </settings:DeviceInformation>
     <Policies>
          <Policy>
               <PolicyType>MS-EAS-Provisioning-WBXML</PolicyType>
               <Status>1</Status>
               <PolicyKey>1307199584</PolicyKey>
               <Data>
                    <EASProvisionDoc>
                         <DevicePasswordEnabled>1</DevicePasswordEnabled>

<AlphanumericDevicePasswordRequired>1</AlphanumericDevicePasswordRequired>
```

```
                        <PasswordRecoveryEnabled>1</PasswordRecoveryEnabled>
                        <RequireStorageCardEncryption>1</RequireStorageCardEncryption>
                        <AttachmentsEnabled>1</AttachmentsEnabled>
                        <MinDevicePasswordLength/>
                        <MaxInactivityTimeDeviceLock>333</MaxInactivityTimeDeviceLock>
                        <MaxDevicePasswordFailedAttempts>8</MaxDevicePasswordFailedAttempts>
                        <MaxAttachmentSize/>
                        <AllowSimpleDevicePassword>0</AllowSimpleDevicePassword>
                        <DevicePasswordExpiration/>
                        <DevicePasswordHistory>0</DevicePasswordHistory>
                    </EASProvisionDoc>
                </Data>
            </Policy>
        </Policies>
    </Provision>
```

4.1.3 Phase 3: Client Acknowledges Receipt and Application of Policy Settings

The client acknowledges the policy download and policy application by using the temporary **PolicyKey** obtained in phase 2. In this case, the client has indicated compliance and provided the correct **PolicyKey**. Therefore, the server responds with the "final" **PolicyKey** which the client then uses in the X-MS-PolicyKey header of successive command requests to satisfy policy enforcement.

Request

```
POST /Microsoft-Server-
ActiveSync?User=deviceuser&DeviceId=6F24CAD599A5BF1A690246B8C68FAE8D&DeviceType=PocketPC&Cmd=
Provision HTTP/1.1
Accept-Language: en-us
MS-ASProtocolVersion: 14.0
Content-Type: application/vnd.ms-sync.wbxml
X-MS-PolicyKey: 1307199584
User-Agent: ASOM
Host: EXCH-B-003

<?xml version="1.0" encoding="utf-8"?>
<Provision xmlns="Provision:">
     <Policies>
          <Policy>
               <PolicyType>MS-EAS-Provisioning-WBXML</PolicyType>
<PolicyKey>1307199584</PolicyKey>
               <Status>1</Status>
          </Policy>
     </Policies>
</Provision>
```

Response

```
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 63
Date: Mon, 01 May 2006 20:15:17 GMT
Content-Type: application/vnd.ms-sync.wbxml
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
MS-Server-ActiveSync: 8.0
Cache-Control: private

<?xml version="1.0" encoding="utf-8"?>
<Provision xmlns="Provision:">
     <Status>1</Status>
```

```
          <Policies>
               <Policy>
                    <PolicyType> MS-EAS-Provisioning-WBXML </PolicyType>
               <Status>1</Status>
               <PolicyKey>3942919513</PolicyKey>
          </Policy>
     </Policies>
</Provision>
```

4.1.4 Phase 4: Client Performs FolderSync by Using the Final PolicyKey

The client uses the "final" policy key obtained in phase 3 in the header of the **FolderSync** command request.

Request

```
POST /Microsoft-Server-
ActiveSync?User=deviceuser&DeviceId=6F24CAD599A5BF1A690246B8C68FAE8D&DeviceType=PocketPC&Cmd=
FolderSync HTTP/1.1
Accept-Language: en-us
MS-ASProtocolVersion: 14.0
Content-Type: application/vnd.ms-sync.wbxml
X-MS-PolicyKey: 3942919513
User-Agent: ASOM
Host: EXCH-B-003

<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
     <SyncKey>0</SyncKey>
</FolderSync>
```

4.2 Directing a Client to Execute a Remote Wipe

The following example shows a set of remote wipe requests and their corresponding responses between a server and a previously provisioned client.

4.2.1 Step 1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=FolderSync&User=T0SyncUser1v14.0&DeviceId=Device1&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
X-MS-PolicyKey: 0
User-Agent: ASOM
Host: EXCH-B-003

<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
     <SyncKey>0</SyncKey>
</FolderSync>
```

4.2.2 Step 1 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 25 Mar 2009 01:23:58 GMT
Content-Length: 15

<?xml version="1.0" encoding="utf-8"?>
<FolderSync >
```

```
     <Status>140</Status>
</FolderSync>
```

4.2.3 Step 2 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Provision&User=T0SyncUser1v14.0&DeviceId=Device1&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
X-MS-PolicyKey: 0
User-Agent: ASOM
Host: EXCH-B-003

<?xml version="1.0" encoding="utf-8"?>
<Provision xmlns="Provision:"></Provision>
```

4.2.4 Step 2 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 25 Mar 2009 01:23:58 GMT
Content-Length: 14

<?xml version="1.0" encoding="utf-8"?>
<Provision>
<Status>1</Status>
<RemoteWipe />
</Provision>
```

4.2.5 Step 3 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Provision&User=T0SyncUser1v14.0&DeviceId=Device1&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
X-MS-PolicyKey: 0
User-Agent: ASOM
Host: EXCH-B-003

<?xml version="1.0" encoding="utf-8"?>
<Provision xmlns="Provision:">
    <RemoteWipe>
        <Status>1</Status>
    </RemoteWipe>
</Provision>
```

4.2.6 Step 3 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 25 Mar 2009 01:24:01 GMT
Content-Length: 14

<?xml version="1.0" encoding="utf-8"?>
<Provision>
<Status>1</Status>
</Provision>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full XML Schema

For ease of implementation, the following sections provide the full **XML schema** for this protocol.

Schema name	Prefix	Section
Provision namespace schema	provision	6.1
Provision request schema	provision	6.2
Provision response schema	provision	6.3

6.1 Provision Namespace Schema

This section contains the contents of the Provision.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining specification
AirSyncBase.xsd	[MS-ASAIRS] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns="Provision" targetNamespace="Provision"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:simpleType name="unsignedByteOrEmpty">
    <xs:union memberTypes="xs:unsignedByte airsyncbase:EmptyTag"/>
  </xs:simpleType>
  <xs:simpleType name="unsignedIntOrEmpty">
    <xs:union memberTypes="xs:unsignedInt airsyncbase:EmptyTag"/>
  </xs:simpleType>
  <xs:element name="PolicyType" type="xs:string"/>
  <xs:element name="PolicyKey" type="xs:string"/>
  <xs:element name="EASProvisionDoc">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DevicePasswordEnabled" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="AlphanumericDevicePasswordRequired" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="PasswordRecoveryEnabled" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="RequireStorageCardEncryption" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="AttachmentsEnabled" type="xs:boolean" minOccurs="0"/>
        <xs:element name="MinDevicePasswordLength" type="unsignedByteOrEmpty"
            minOccurs="0"/>
        <xs:element name="MaxInactivityTimeDeviceLock"
            type="unsignedIntOrEmpty" minOccurs="0"/>
        <xs:element name="MaxDevicePasswordFailedAttempts"
            type="unsignedByteOrEmpty" minOccurs="0"/>
        <xs:element name="MaxAttachmentSize" type="unsignedIntOrEmpty"
            minOccurs="0"/>
        <xs:element name="AllowSimpleDevicePassword" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="DevicePasswordExpiration" type="unsignedIntOrEmpty"
            minOccurs="0"/>
        <xs:element name="DevicePasswordHistory" type="xs:unsignedInt"
            minOccurs="0"/>
        <xs:element name="AllowStorageCard" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowCamera" type="xs:boolean" minOccurs="0"/>
```

```
        <xs:element name="RequireDeviceEncryption" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="AllowUnsignedApplications" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="AllowUnsignedInstallationPackages" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="MinDevicePasswordComplexCharacters"
            type="xs:unsignedByte" minOccurs="0"/>
        <xs:element name="AllowWiFi" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowTextMessaging" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowPOPIMAPEmail" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowBluetooth" type="xs:unsignedByte"
            minOccurs="0"/>
        <xs:element name="AllowIrDA" type="xs:boolean" minOccurs="0"/>
        <xs:element name="RequireManualSyncWhenRoaming" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="AllowDesktopSync" type="xs:boolean" minOccurs="0"/>
        <xs:element name="MaxCalendarAgeFilter" type="xs:unsignedInt"
            minOccurs="0"/>
        <xs:element name="AllowHTMLEmail" type="xs:boolean" minOccurs="0"/>
        <xs:element name="MaxEmailAgeFilter" type="xs:unsignedInt"
            minOccurs="0"/>
        <xs:element name="MaxEmailBodyTruncationSize" type="xs:integer"
            minOccurs="0"/>
        <xs:element name="MaxEmailHTMLBodyTruncationSize" type="xs:integer"
            minOccurs="0"/>
        <xs:element name="RequireSignedSMIMEMessages" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="RequireEncryptedSMIMEMessages" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="RequireSignedSMIMEAlgorithm" type="xs:integer"
            minOccurs="0"/>
        <xs:element name="RequireEncryptionSMIMEAlgorithm" type="xs:integer"
            minOccurs="0"/>
        <xs:element name="AllowSMIMEEncryptionAlgorithmNegotiation"
            type="xs:integer" minOccurs="0"/>
        <xs:element name="AllowSMIMESoftCerts" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="AllowBrowser" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowConsumerEmail" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowRemoteDesktop" type="xs:boolean" minOccurs="0"/>
        <xs:element name="AllowInternetSharing" type="xs:boolean"
            minOccurs="0"/>
        <xs:element name="UnapprovedInROMApplicationList" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ApplicationName" type="xs:string" minOccurs="0"
                  maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="ApprovedApplicationList" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Hash" type="xs:string" minOccurs="0"
                  maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.2 Provision Request Schema

This section contains the contents of the ProvisionRequest.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
Provision.xsd	6.1
SettingsRequest.xsd	[MS-ASCMD] section 6.39

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:settings=
    "Settings" xmlns="Provision" targetNamespace="Provision"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="Provision.xsd"/>
  <xs:import namespace="Settings" schemaLocation="SettingsRequest.xsd"/>
  <xs:element name="Provision">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="settings:DeviceInformation" minOccurs="0"/>
        <xs:element name="Policies" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Policy">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="PolicyType"/>
                    <xs:element ref="PolicyKey" minOccurs="0"/>
                    <xs:element name="Status" type="xs:string" minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="RemoteWipe" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Status">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:minInclusive value="1"/>
                    <xs:maxInclusive value="2"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="AccountOnlyRemoteWipe" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Status">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:minInclusive value="1"/>
                    <xs:maxInclusive value="2"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```
</xs:schema>
```

6.3 Provision Response Schema

This section contains the contents of the ProvisionResponse.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
Provision.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
SettingsResponse.xsd	[MS-ASCMD] section 6.40

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:settings="Settings" xmlns="Provision"
    targetNamespace="Provision" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="Provision.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="Settings" schemaLocation="SettingsResponse.xsd"/>
  <xs:element name="Provision">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="settings:DeviceInformation" minOccurs="0"/>
        <xs:element name="Status" type="xs:unsignedByte"/>
        <xs:element name="Policies" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Policy">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="PolicyType"/>
                    <xs:element name="Status" type="xs:unsignedByte"/>
                    <xs:element ref="PolicyKey" minOccurs="0"/>
                    <xs:element name="Data" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element ref="EASProvisionDoc"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="RemoteWipe" type="airsyncbase:EmptyTag"
            minOccurs="0"/>
        <xs:element name="AccountOnlyRemoteWipe" type="airsyncbase:EmptyTag"
            minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Exchange Server 2007 Service Pack 1 (SP1)
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019
- Windows 8.1 operating system
- Windows 10 operating system
- Windows Server 2016 operating system
- Windows 11 operating system

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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.

<1> Section 3.1.5.1: Windows 8.1 does not send a **Provision** command when contacting the server for the first time.

<2> Section 3.1.5.2: In Microsoft Exchange Server 2007 and Exchange 2010, this does not cause status code 139.

<3> Section 3.2.5.1.2.2: In Exchange 2007 and Exchange 2010, if the client reports success, the server returns a value of 1 in the **Status** element and a remote wipe directive.

<4> Section 3.2.5.1.2.2: In Exchange 2007 and Exchange 2010, if the client reports failure, the server returns a value of 1 in the **Status** element.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
7 Appendix B: Product Behavior	Updated list of supported products.	Major

9 Index

V

X