[MS-ASPROV]:
ActiveSync Provisioning Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0.0	Major	Initial Release.
03/04/2009	1.0.1	Editorial	Revised and edited technical content.
04/10/2009	2.0.0	Major	Updated technical content and applicable product releases.
07/15/2009	3.0.0	Major	Revised and edited for technical content.
11/04/2009	3.1.0	Minor	Updated the technical content.
02/10/2010	3.1.0	None	Version 3.1.0 Release
05/05/2010	4.0.0	Major	Updated and revised the technical content.
08/04/2010	5.0	Major	Significantly changed the technical content.
11/03/2010	5.1	Minor	Clarified the meaning of the technical content.
03/18/2011	6.0	Major	Significantly changed the technical content.
08/05/2011	6.1	Minor	Clarified the meaning of the technical content.
10/07/2011	6.2	Minor	Clarified the meaning of the technical content.
01/20/2012	7.0	Major	Significantly changed the technical content.
04/27/2012	7.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

The 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 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Hypertext Transfer Protocol (HTTP)
XML

The following terms are defined in [MS-OXGLOS]:

base64 encoding
cabinet (.cab) file
encrypted message
header
Hypertext Markup Language (HTML)
permission
plain text
Short Message Service (SMS)
Uniform Resource Identifier (URI)
Wireless Application Protocol (WAP) Binary XML (WBXML)
XML namespace
XML schema

The following terms are specific to this document:

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.

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

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. Please check the archive site,

http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

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

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

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

[MS-ASWBXML] Microsoft Corporation, "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, http://www.rfc-editor.org/rfc/rfc2119.txt

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

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

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary".

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].

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 that are embedded within a request or response that is associated with the **Provision** command ([MS-ASCMD] section 2.2.2.12).

2.2 Message Syntax

The XML markup that constitutes the Request Body or the Response Body is transmitted between client and server by using **Wireless Application Protocol (WAP) Binary XML (WBXML)**. For details, see [MS-ASWBXML].

The following is the XML schema definition for the protocol request, defined in accordance with the rules specified in [XMLSCHEMA1].

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Provision:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Provision:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:settings="Settings:">

    <xs:import namespace="Settings:"/>

    <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 name="PolicyType" type="xs:string" />
                                        <xs:element name="PolicyKey" type="xs:string"
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" type="xs:string" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
```

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

The following is the XML schema definition for the protocol response.

```
<?xml version="1.0" ?>
<xs:schema
    xmlns:tns="Provision:"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    targetNamespace="Provision:"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="Provision">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Status" type="xs:unsignedByte" />
             <xs:element name="Policies">
                 <xs:complexType>
                     <xs:sequence>
                         <xs:element name="Policy">
                             <xs:complexType>
                                 <xs:sequence>
                                     <xs:element name="PolicyType" type="xs:string" />
                                     <xs:element name="Status" type="xs:unsignedByte" />
                                     <xs:element name="PolicyKey" type="xs:string" />
                                     <xs:element name="Data">
                                         <xs:complexType>
                                           <xs:sequence>
                                             <xs:element name="EASProvisionDoc">
                                               <xs:complexType>
                                                 <xs:sequence>
                                                   <xs:element name="DevicePasswordEnabled"
minOccurs="0" type="xs:boolean" />
                                                   <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="xs:unsignedByte" minOccurs="0" nillable="true" />
                                                   <xs:element
name="MaxInactivityTimeDeviceLock" type="xs:unsignedInt" minOccurs="0" nillable="true" />
                                                   <xs:element
name="MaxDevicePasswordFailedAttempts" type="xs:unsignedByte" minOccurs="0" nillable="true"
/>
                                                   <xs:element name="MaxAttachmentSize"
type="xs:unsignedInt" minOccurs="0" nillable="true" />
                                                   <xs:element
name="AllowSimpleDevicePassword" type="xs:boolean" minOccurs="0" />
                                                   <xs:element
name="DevicePasswordExpiration" type="xs:unsignedInt" minOccurs="0" nillable="true" />
                                                   <xs:element name="DevicePasswordHistory"
type="xs:unsignedint" minOccurs="0" nillable="true" />
                                                   <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"
nillable="true" />
                                                <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" nillable="true" />
                                                <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" nillable="true" />
                                                <xs:element name="AllowHTMLEmail"
type="xs:boolean" minOccurs="0" />
                                                <xs:element name="MaxEmailAgeFilter"
type="xs:unsignedInt" minOccurs="0" nillable="true" />
                                                <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" minOccurs="0" type="xs:string" />
                                                        </xs:sequence>
```

```
                                                    </xs:complexType>
                                                </xs:element>
                                                <xs:element name="ApprovedApplicationList"
minOccurs="0">
                                                    <xs:complexType>
                                                        <xs:sequence>
                                                            <xs:element name="Hash"
minOccurs="0" type="xs:string" />
                                                        </xs:sequence>
                                                    </xs:complexType>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:schema>
```

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 2.2.2.2, 2.2.2.3, 2.2.2.4, 2.2.2.5
settings	Settings	[MS-ASCMD] section 2.2.2.16
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
AllowBluetooth (section 2.2.2.1)	Whether Bluetooth and hands-free profiles are allowed on the device.
AllowBrowser (section 2.2.2.2)	Whether the device allows the use of a Web browser.
AllowCamera (section 2.2.2.3)	Whether the device allows the use of the built-in camera.
AllowConsumerEmail (section 2.2.2.4)	Whether the device allows the use of personal e-mail.
AllowDesktopSync (section 2.2.2.5)	Whether the device allows synchronization with Desktop ActiveSync.
AllowHTMLEmail (section 2.2.2.6)	Whether the device uses **HTML**-formatted e-mail.
AllowInternetSharing (section 2.2.2.7)	Whether the device allows the use of Internet Sharing.
AllowIrDA (section 2.2.2.8)	Whether the device allows the use of IrDA (infrared) connections.
AllowPOPIMAPEmail (section 2.2.2.9)	Whether the device allows access to POP/IMAP e-mail.
AllowRemoteDesktop (section 2.2.2.10)	Whether the device allows the use of Remote Desktop.
AllowSimpleDevicePassword (section 2.2.2.11)	Whether the device allows simple passwords.
AllowSMIMEEncryptionAlgorithmNegotiation (section 2.2.2.12)	Whether the device can negotiate the encryption algorithm to be used for signing.
AllowSMIMESoftCerts (section 2.2.2.13)	Whether the device uses soft certificates to sign outgoing messages.
AllowStorageCard (section 2.2.2.14)	Whether the device allows the use of the storage card.
AllowTextMessaging (section 2.2.2.15)	Whether the device allows **Short Message Service (SMS)**/text messaging.
AllowUnsignedApplications (section 2.2.2.16)	Whether the device allows unsigned applications to execute.
AllowUnsignedInstallationPackages (section 2.2.2.17)	Whether the device allows unsigned **cabinet (.cab) files** to be installed.
AllowWiFi (section 2.2.2.18)	Whether the device allows the use of Wi-Fi connections.
AlphanumericDevicePasswordRequired (section 2.2.2.19)	Indicates whether a client device requires an alphanumeric password.
ApplicationName (section 2.2.2.20)	The name of an in-ROM application (.exe file) that is not approved for execution.

Element name	Description
ApprovedApplicationList (section 2.2.2.21)	A list of in-RAM applications that are approved for execution.
AttachmentsEnabled (section 2.2.2.22)	Indicates whether e-mail attachments are enabled.
Data (section 2.2.2.23)	The settings for a policy.
DevicePasswordEnabled (section 2.2.2.24)	Indicates whether a client device requires a password.
DevicePasswordExpiration (section 2.2.2.25)	Whether the password expires after the specified number of days, as determined by the policy.
DevicePasswordHistory (section 2.2.2.26)	The minimum number of previously used passwords the client device stores to prevent reuse.
EASProvisionDoc (section 2.2.2.27)	The collection of security settings for device provisioning.
Hash (section 2.2.2.28)	The SHA-1 hash of an in-memory application that is approved for execution.
MaxAttachmentSize (section 2.2.2.29)	The maximum attachment size, as determined by the security policy.
MaxCalendarAgeFilter (section 2.2.2.30)	The maximum number of calendar days that can be synchronized.
MaxDevicePasswordFailedAttempts (section 2.2.2.31)	The number of password failures that are permitted before the device is wiped.
MaxEmailAgeFilter (section 2.2.2.32)	The e-mail age limit for synchronization.
MaxEmailBodyTruncationSize (section 2.2.2.33)	The truncation size for **plain text**–formatted e-mail messages.
MaxEmailHTMLBodyTruncationSize (section 2.2.2.34)	The truncation size for HTML-formatted e-mail messages.
MaxInactivityTimeDeviceLock (section 2.2.2.35)	The number of seconds of inactivity before the device locks itself.
MinDevicePasswordComplexCharacters (section 2.2.2.36)	The minimum number of complex characters (numbers and symbols) contained within the password.
MinDevicePasswordLength (section 2.2.2.37)	The minimum device password length that the user can enter.
PasswordRecoveryEnabled (section 2.2.2.38)	Indicates whether to enable a recovery password to be sent to the server by using the **Settings** command.
Policies (section 2.2.2.39)	A collection of security policies.
Policy (section 2.2.2.40)	A policy.

Element name	Description
PolicyKey (section 2.2.2.41)	Used by the server to mark the state of policy settings on the client.
PolicyType (section 2.2.2.42)	Specifies the format in which the policy settings are to be provided.
Provision (section 2.2.2.43)	The capabilities and permissions for the device.
RemoteWipe (section 2.2.2.44)	Specifies either a **remote wipe** directive from the server or a client's confirmation of a remote wipe directive.
RequireDeviceEncryption (section 2.2.2.45)	Whether the device uses encryption.
RequireEncryptedSMIMEMessages (section 2.2.2.46)	Whether the device is required to send **encrypted messages**.
RequireEncryptionSMIMEAlgorithm (section 2.2.2.47)	The algorithm to be used when encrypting a message.
RequireManualSyncWhenRoaming (section 2.2.2.48)	Whether the device requires manual synchronization when the device is roaming.
RequireSignedSMIMEAlgorithm (section 2.2.2.49)	The algorithm to be used when signing a message.
RequireSignedSMIMEMessages (section 2.2.2.50)	Whether the device is required to send signed S/MIME messages.
RequireStorageCardEncryption (section 2.2.2.51)	Indicates whether the device has to encrypt content that is stored on the storage card.
settings:DeviceInformation (section 2.2.2.52)	Specifies the settings for the device in an initial Provisioning request.
Status (section 2.2.2.53)	Indicates success or failure of specific parts of a command.
UnapprovedInROMApplicationList (section 2.2.2.54)	A list of in-ROM applications that are not approved for execution.

2.2.2.1 AllowBluetooth

The **AllowBluetooth** element is an optional child element of type **unsignedByte** ([MS-ASDTYPE] section 2.7) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

Value	Meaning
2	Allow Bluetooth.

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

2.2.2.2 AllowBrowser

The **AllowBrowser** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.3 AllowCamera

The **AllowCamera** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.4 AllowConsumerEmail

The **AllowConsumerEmail** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device allows the user to configure a personal e-mail 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 e-mail account.

Value	Meaning
1	Allow the user to configure a personal e-mail account.

2.2.2.5 AllowDesktopSync

The **AllowDesktopSync** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.6 AllowHTMLEmail

The **AllowHTMLEmail** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device uses HTML-formatted e-mail.

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

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

Value	Meaning
0	HTML-formatted e-mail is not allowed.
1	HTML-formatted e-mail is allowed.

2.2.2.7 AllowInternetSharing

The **AllowInternetSharing** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.8 AllowIrDA

The **AllowIrDA** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.9 AllowPOPIMAPEmail

The **AllowPOPIMAPEmail** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device allows access to POP or IMAP e-mail.

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

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

Value	Meaning
0	POP or IMAP e-mail access is not allowed.
1	POP or IMAP e-mail access is allowed.

2.2.2.10 AllowRemoteDesktop

The **AllowRemoteDesktop** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.11 AllowSimpleDevicePassword

The **AllowSimpleDevicePassword** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device 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 2.2.2.24) is set to FALSE (0), the client SHOULD ignore this element.

2.2.2.12 AllowSMIMEEncryptionAlgorithmNegotiation

The **AllowSMIMEEncryptionAlgorithmNegotiation** element is an optional child element of type **integer** ([MS-ASDTYPE] section 2.5) of the **EASProvisionDoc** element (section 2.2.2.12) 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.

2.2.2.13 AllowSMIMESoftCerts

The **AllowSMIMESoftCerts** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device 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.

Value	Meaning
1	Soft certificates are allowed.

2.2.2.14 AllowStorageCard

The **AllowStorageCard** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.15 AllowTextMessaging

The **AllowTextMessaging** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.16 AllowUnsignedApplications

The **AllowUnsignedApplications** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.17 AllowUnsignedInstallationPackages

The **AllowUnsignedInstallationPackages** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.18 AllowWifi

The **AllowWifi** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.19 AlphanumericDevicePasswordRequired

The **AlphanumericDevicePasswordRequired** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether a device 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 2.2.2.24) is FALSE (0), the client ignores this element.

2.2.2.20 ApplicationName

The **ApplicationName** element is an optional child element of type **string** ([MS-ASDTYPE] section 2.6) of the **UnapprovedInROMApplicationList** element (section 2.2.2.54) 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.

The **UnapprovedInROMApplicationList** element has at least one instance of the **ApplicationName** element.

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

2.2.2.21 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 2.2.2.27). 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 2.2.2.28): This element is optional.

2.2.2.22 AttachmentsEnabled

The **AttachmentsEnabled** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether e-mail attachments are enabled.

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

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

Value	Meaning
0	Attachments are not allowed.
1	Attachments are allowed.

2.2.2.23 Data

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

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

- **EASProvisionDoc** (section 2.2.2.27): One instance of this element is required.

2.2.2.24 DevicePasswordEnabled

The **DevicePasswordEnabled** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether a device 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.

2.2.2.25 DevicePasswordExpiration

The **DevicePasswordExpiration** element is an optional child element of type **unsignedInt** of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the maximum number of days until a password expires.

The **DevicePasswordExpiration** element can be empty, indicating that no device 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 2.2.2.24) is set to FALSE (0), the client SHOULD ignore this element.

2.2.2.26 DevicePasswordHistory

The **DevicePasswordHistory** element is an optional child element of type **unsignedInt** of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the minimum number of previously used passwords stored to prevent reuse on the client device.

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.

Value	Meaning
>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 2.2.2.24) 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.

2.2.2.27 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 2.2.2.23).

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

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

- **AllowBluetooth** (section 2.2.2.1)
- **AllowBrowser** (section 2.2.2.2)
- **AllowCamera** (section 2.2.2.3)
- **AllowConsumerEmail** (section 2.2.2.4)
- **AllowDesktopSync** (section 2.2.2.5)
- **AllowHTMLEmail** (section 2.2.2.6)
- **AllowInternetSharing** (section 2.2.2.7)
- **AllowIrDA** (section 2.2.2.8)
- **AllowPOPIMAPEmail** (section 2.2.2.9)
- **AllowRemoteDesktop** (section 2.2.2.10)
- **AllowSimpleDevicePassword** (section 2.2.2.11)
- **AllowSMIMEEncryptionAlgorithmNegotiation** (section 2.2.2.12)
- **AllowSMIMESoftCerts** (section 2.2.2.13)
- **AllowStorageCard** (section 2.2.2.14)
- **AllowTextMessaging** (section 2.2.2.15)
- **AllowUnsignedApplications** (section 2.2.2.16)
- **AllowUnsignedInstallationPackages** (section 2.2.2.17)

- **AllowWifi** (section 2.2.2.18)
- **AlphanumericDevicePasswordRequired** (section 2.2.2.19)
- **ApprovedApplicationList** (section 2.2.2.21)
- **AttachmentsEnabled** (section 2.2.2.22)
- **DevicePasswordEnabled** (section 2.2.2.24)
- **DevicePasswordExpiration** (section 2.2.2.25)
- **DevicePasswordHistory** (section 2.2.2.26)
- **MaxAttachmentSize** (section 2.2.2.29)
- **MaxCalendarAgeFilter** (section 2.2.2.30)
- **MaxDevicePasswordFailedAttempts** (section 2.2.2.31)
- **MaxEmailAgeFilter** (section 2.2.2.32)
- **MaxEmailBodyTruncationSize** (section 2.2.2.33)
- **MaxEmailHTMLBodyTruncationSize** (section 2.2.2.34)
- **MaxInactivityTimeDeviceLock** (section 2.2.2.35)
- **MinDevicePasswordComplexCharacters** (section 2.2.2.36)
- **MinDevicePasswordLength** (section 2.2.2.37)
- **PasswordRecoveryEnabled** (section 2.2.2.38)
- **RequireDeviceEncryption** (section 2.2.2.45)
- **RequireEncryptedSMIMEMessages** (section 2.2.2.46)
- **RequireEncryptionSMIMEAlgorithm** (section 2.2.2.47)
- **RequireManualSyncWhenRoaming** (section 2.2.2.48)
- **RequireSignedSMIMEAlgorithm** (section 2.2.2.49)
- **RequireSignedSMIMEMessages** (section 2.2.2.50)
- **RequireStorageCardEncryption** (section 2.2.2.51)
- **UnapprovedInROMApplicationList** (section 2.2.2.54)

2.2.2.28 Hash

The **Hash** element is an optional child element of type **string** ([MS-ASDTYPE] section 2.6) of the **ApprovedApplicationList** element (section 2.2.2.21) 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.

2.2.2.29 MaxAttachmentSize

The **MaxAttachmentSize** element is an optional child element of type **unsignedInt** of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the maximum attachment size in bytes as determined by security policy.

The **EASProvisionDoc** element has one instance of the **MaxAttachmentSize** element.

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

2.2.2.30 MaxCalendarAgeFilter

The **MaxCalendarAgeFilter** element is an optional child element of type **unsignedInt** of the **EASProvisionDoc** element (section 2.2.2.27) 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

2.2.2.31 MaxDevicePasswordFailedAttempts

The **MaxDevicePasswordFailedAttempts** element is an optional child element of type **unsignedByte** ([MS-ASDTYPE] section 2.7) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the maximum number of failed password logon attempts that are permitted before the device SHOULD perform a local wipe or enter a timed lock out mode.

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

Valid values for **MaxDevicePasswordFailedAttempts** are in the range from 4 through 16.

If the **MaxDevicePasswordFailedAttempts** element is included in a response, and the value of the **DevicePasswordEnabled** element (section 2.2.2.24) is set to FALSE (0), the client ignores this element.

2.2.2.32 MaxEmailAgeFilter

The **MaxEmailAgeFilter** element is an optional child element of type **unsignedInt** of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the e-mail 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 e-mail.

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

2.2.2.33 MaxEmailBodyTruncationSize

The **MaxEmailBodyTruncationSize** element is an optional child element of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the truncation size for plain text–formatted e-mail.

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

Valid values for the **MaxEmailBodyTruncationSize** element are an **integer** ([MS-ASDTYPE] section 2.5) 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 e-mail body to the specified size.

2.2.2.34 MaxEmailHTMLBodyTruncationSize

The **MaxEmailHTMLBodyTruncationSize** element is an optional child element of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the truncation size for HTML-formatted e-mail.

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

Valid values for the **MaxEmailHTMLBodyTruncationSize** element are an **integer** ([MS-ASDTYPE] section 2.5) 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 e-mail body to the specified size.

2.2.2.35 MaxInactivityTimeDeviceLock

The **MaxInactivityTimeDeviceLock** element is an optional child element of type **unsignedInt** of the **EASProvisionDoc** element (section 2.2.2.27) 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 not included in a response, the client interprets this as meaning that no time device lock has been set by the security policy.

2.2.2.36 MinDevicePasswordComplexCharacters

The **MinDevicePasswordComplexCharacters** element is an optional child element of type **unsignedByte** ([MS-ASDTYPE] section 2.7) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the required level of complexity of the device 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.

2.2.2.37 MinDevicePasswordLength

The **MinDevicePasswordLength** element is an optional child element of type **unsignedByte** ([MS-ASDTYPE] section 2.7) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies the minimum device password length that the user can enter.

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

MinDevicePasswordLength MUST have a value no less than 1 and no greater than 16. If 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 2.2.2.24) is FALSE (0), the client SHOULD ignore this element.

2.2.2.38 PasswordRecoveryEnabled

The **PasswordRecoveryEnabled** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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 device 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 device 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 2.2.2.24) is FALSE (0), the client SHOULD ignore this element. This element SHOULD be ignored if the client does not support recovery passwords.

2.2.2.39 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 2.2.2.43).

A command response has one top-level **Policies** element per response.

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

- **Policy** (section 2.2.2.40): At least one element of this type is required.

2.2.2.40 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 2.2.2.39).

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 2.2.2.42) (required)

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

- **PolicyType** (section 2.2.2.42) (required)
- **PolicyKey** (section 2.2.2.41) (required)
- **Status** (section 2.2.2.53) (required)
- **Data** (section 2.2.2.23) (required)

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

- **PolicyType** (section 2.2.2.42) (required)
- **PolicyKey** (section 2.2.2.41) (required, MUST appear before the **Status** element)

- **Status** (section 2.2.2.53) (required)

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

- **PolicyType** (section 2.2.2.42) (required)
- **PolicyKey** (section 2.2.2.41) (required)
- **Status** (section 2.2.2.53) (required)

2.2.2.41 PolicyKey

The **PolicyKey** element is an optional element of type **string** ([MS-ASDTYPE] section 2.6) with a maximum of 64 characters and no child elements. It is a child element of the **Policy** element (section 2.2.2.40).

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.

2.2.2.42 PolicyType

The **PolicyType** element is a child element of type **string** ([MS-ASDTYPE] section 2.6) of the **Policy** element (section 2.2.2.40) that, in the download policy settings phase, specifies the format in which the policy settings are to be provided to the client device.

PolicyType MUST be "MS-EAS-Provisioning-WBXML".

2.2.2.43 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 2.2.2.52)
- **Status** (section 2.2.2.53)
- **Policies** (section 2.2.2.39)
- **RemoteWipe** (section 2.2.2.44)

2.2.2.44 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 by the client only in response to a remote wipe directive from the server.

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

- **Status** (section 2.2.2.53): One element of this type is required in a remote wipe client request.

2.2.2.45 RequireDeviceEncryption

The **RequireDeviceEncryption** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device 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.

2.2.2.46 RequireEncryptedSMIMEMessages

The **RequireEncryptedSMIMEMessages** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device sends encrypted e-mail messages.

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

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

Value	Meaning
0	Encrypted e-mail messages are not required.
1	E-mail messages are required to be encrypted.

2.2.2.47 RequireEncryptionSMIMEAlgorithm

The **RequireEncryptionSMIMEAlgorithm** element is an optional child element of type **integer** ([MS-ASDTYPE] section 2.5) of the **EASProvisionDoc** element (section 2.2.2.27) 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

Value	Meaning
4	RC240bit

2.2.2.48 RequireManualSyncWhenRoaming

The **RequireManualSyncWhenRoaming** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.49 RequireSignedSMIMEAlgorithm

The **RequireSignedSMIMEAlgorithm** element is an optional child element of type **integer** ([MS-ASDTYPE] section 2.5) of the **EASProvisionDoc** element (section 2.2.2.27) 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.

2.2.2.50 RequireSignedSMIMEMessages

The **RequireSignedSMIMEMessages** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device 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.

2.2.2.51 RequireStorageCardEncryption

The **RequireStorageCardEncryption** element is an optional child element of type **boolean** ([MS-ASDTYPE] section 2.1) of the **EASProvisionDoc** element (section 2.2.2.27) that specifies whether the device encrypts content that is stored on the device.

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.

2.2.2.52 settings:DeviceInformation

The **settings:DeviceInformation** element is an optional<1> **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 2.2.2.43). The **settings:DeviceInformation** element is defined in the Settings XML namespace, as specified in [MS-ASCMD] section 2.2.2.16.

A client SHOULD send the **settings:DeviceInformation** element with its contents when sending an initial **Provision** command request to the server but not on subsequent requests.<2>

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 2.2.2.16.

2.2.2.53 Status

The **Status** element is a child element of the **Policy** element (section 2.2.2.40), the **Provision** element (section 2.2.2.43), and the **RemoteWipe** element (section 2.2.2.44). The definition of this element differs according to the context in which it is used. For more details, see section 2.2.2.53.1, section 2.2.2.53.2, and section 2.2.2.53.3.

2.2.2.53.1 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 2.7). 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.

Value	Meaning
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 2.6). 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.

2.2.2.53.2 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 2.7). The value indicates the success or failure of the **Provision** command. The following table lists valid values for the **Status** element when it is the child of the **Provision** element.

Value	Meaning
1	Success
2	Protocol error
3	General server error
4	The device is externally managed

2.2.2.53.3 Status (RemoteWipe)

The **Status** element is a required child of the **RemoteWipe** element in command requests. The value of this element is a **string** ([MS-ASDTYPE] section 2.6). 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** element.

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

2.2.2.54 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 2.2.2.27). 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 2.2.2.20): This element is optional.

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.

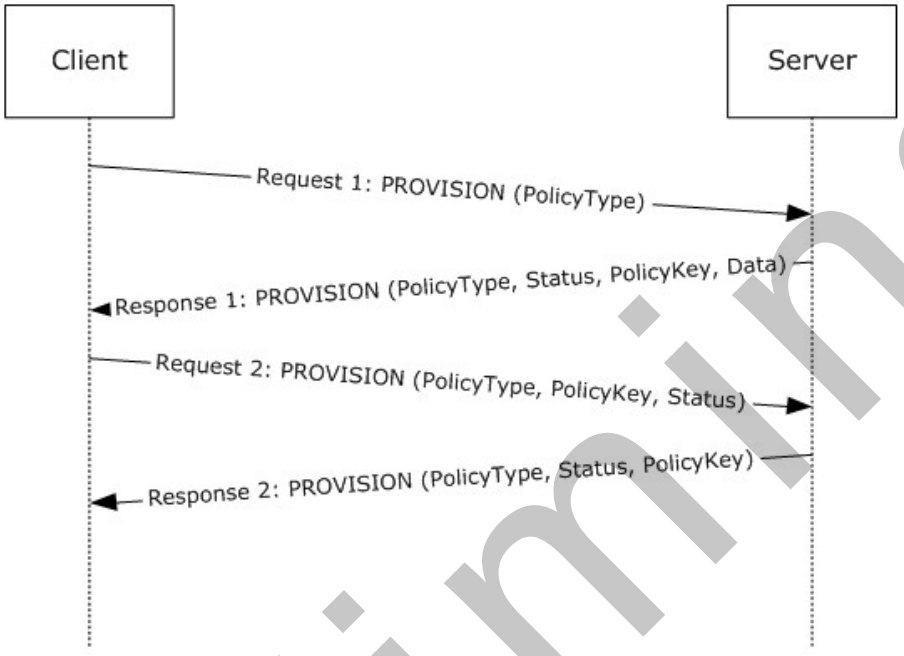

Figure 1: Downloading policy settings

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 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 begin the provisioning process in the following scenarios:

- When contacting the server for the first time.
- When the server returns a status code<3> from any command indicating that the client needs to re-provision.
- When the server returns a status code from any command requesting a remote wipe.

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.

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 2.2.1.1.2.6) 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 for the **Autodiscover** command ([MS-ASCMD] section 2.2.2.1), the **Ping** command ([MS-ASCMD] section 2.2.2.11), and the **HTTP OPTIONS** command ([MS-ASHTTP] section 2.2.3).

3.1.5.1.1 Initial Request

During the initial request, the current policy key MUST be reset to 0 (zero).

To request security policy settings, the initial provisioning request uses the following format.

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

All of these elements are required except for the **settings:DeviceInformation** element (specified in [MS-ASCMD] section 2.2.3.45). However, clients SHOULD<4> include a **settings:DeviceInformation** element within the **Provision** element.

If the initial provisioning request is in response to receiving a status code from the server indicating that a remote wipe is requested, the request SHOULD consist of an empty **Provision** element.

If the server response contains a **RemoteWipe** (section 2.2.2.44) element within the **Provision** element, the client SHOULD acknowledge the remote wipe, as specified in section 3.1.5.1.2.2. 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 2.2.2.53.2) within the **Provision** element that indicates success, and also contains a **Policies** element within the **Provision** element, the client ensures that the security policy settings contained in the **Policy** element that has a **PolicyType** child element with a value of "MS-EAS-Provisioning-WBXML" are actually enforced, and acknowledges the security policy settings, as specified in section 3.1.5.1.2.1. The value of the **PolicyKey** element contained within this **Policy** element is a temporary policy key that is only valid for the acknowledgment request.

Any **Policy** elements that have a value for their **PolicyType** child element other than "MS-EAS-Provisioning-WBXML" SHOULD be ignored.

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 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 2.2.2.53.1.

If the server response includes a **Status** element (section 2.2.2.53.2) within the **Provision** element that indicates success, and also contains a **Policies** element within the **Provision** element, the client checks for a **Policy** element that has a **PolicyType** child element with a value of "MS-EAS-Provisioning-WBXML". The value of the **PolicyKey** element contained within this **Policy** element is a permanent policy key that is valid for subsequent command requests.

Any **Policy** elements that have a value for their **PolicyType** child element other than "MS-EAS-Provisioning-WBXML" SHOULD be ignored.

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 2.2.2.44) 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 2.2.2.53.3) to indicate the result of the 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. 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.

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.

Code	Meaning	Cause	Resolution
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. The server is configured to not allow clients that cannot fully apply the policy.	Server administrator intervention is required.
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 2.2.1.1.2.6) 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 devices.	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.

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 a value of "MS-EAS-Provisioning-WBXML" in the **PolicyType** element.
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 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.

The current security policy settings on the client are represented by the current policy key, which SHOULD be received from the client in the **X-MS-PolicyKey** header ([MS-ASHTTP] section 2.2.1.1.2.6) 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 current policy key is received from the client for all protocol command requests except for the **Autodiscover**command ([MS-

ASCMD] section 2.2.2.1), the **Ping** command ([MS-ASCMD] section 2.2.2.11), and the **HTTP OPTIONS** command ([MS-ASHTTP] section 2.2.3). The server generates, stores, and sends the policy key when it responds to a **Provision** command request for policy settings. If the policy key sent by the client does not match the stored policy key, or if the server determines that policy settings need to be updated on the client, the server SHOULD<5> return a status code 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 sent by the client device in a **settings:DeviceInformation** element ([MS-ASCMD] section 2.2.3.45) and SHOULD respond to a security policy settings request in an initial **Provision** command request with a response in the following format.

```
<Provision>
   <settings:DeviceInformation>
      <settings:Status>...</settings:Status>
   </settings:DeviceInformation>
   <Status>...</Status>
   <Policies>
      <Policy>
         <PolicyType>MS-EAS-Provisioning-WBXML</PolicyType>
         <Status>...</Status>
         <PolicyKey>...</PolicyKey>
         <Data>
            <EASProvisionDoc>
               ...
            </EASProvisionDoc>
         </Data>
      </Policy>
   </Policies>
</Provision>
```

The value of the **PolicyKey** element (section 2.2.2.41) is a temporary policy key that will only be valid for an acknowledgment request to acknowledge the policy settings contained in the **EASProvisionDoc** element (section 2.2.2.27).

The **EASProvisionDoc** element contains child elements that represent the policy settings configured for the user account. The server MAY specify any of the child elements of the **EASProvisionDoc** element as empty tags. In these cases, the client can unset these values if they were previously set, but SHOULD NOT otherwise modify the current value of that element.

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

```
<Provision>
   <Status>...</Status>
   <RemoteWipe/>
</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 device. If it does not, the server SHOULD return a **Status** (section 2.2.2.53.2) 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 2.2.2.53.1) 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 2.2.2.53.2) 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>MS-EAS-Provisioning-WBXML</PolicyType>
         <Status>...</Status>
         <PolicyKey>...</PolicyKey>
      </Policy>
   </Policies>
</Provision>
```

The value of the **PolicyKey** element (section 2.2.2.41) 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 2.2.2.53.3) of the acknowledgment request. If the client reports success, the server SHOULD return a value of 1 in the **Status** element (section 2.2.2.53.2). If the client reports failure, the server SHOULD return a value of 2 in the **Status** element and should respond to the next command request from the client with a remote wipe directive.

The server's response is in the following format.

```
<Provision>
   <Status>...</Status>
   <RemoteWipe/>
</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 by using MS-EAS-Provisioning-WBXML.
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<6> because the server has determined that the device 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 2.2.2.41). 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 2.2.2.52) 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: Product Behavior

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

- Microsoft® Exchange Server 2007 Service Pack 1 (SP1)
- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 15 Technical Preview

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

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

<1> Section 2.2.2.52: When the MS-ASProtocolVersion header value is 14.1, the **settings:DeviceInformation** element is required. The **settings:DeviceInformation** element MUST contain a **Set** ([MS-ASCMD] section 2.2.3.152) child element, and the **Set** child element MUST contain a **Model** ([MS-ASCMD] section 2.2.3.105) child element.

<2> Section 2.2.2.52: When the MS-ASProtocolVersion header value is 14.0, clients send <settings:DeviceInformation> parameters using the **Settings** command to the server as soon as possible after the client has been provisioned and before the **FolderSync** command so that the server can use this information to determine what the device has access to.

<3> Section 3.1.5.1: When the MS-ASProtocolVersion header is set to 12.1, the server sends an HTTP 449 response to request a **Provision** command from the client.

<4> Section 3.1.5.1.1: The **settings:DeviceInformation** element in the **Provision** command is not supported from the client when the MS-ASProtocolVersion header is set to 12.1. In this case, the client uses the **Settings** command to receive this information.

<5> Section 3.2.5.1: When the MS-ASProtocolVersion header is set to 12.1, the server sends an HTTP 449 response to indicate that the client needs to request the security policy settings and obtain a new policy key.

<6> Section 4.1.1: When the MS-ASProtocolVersion header is set to 12.1, the server returns status code HTTP 449.

7 Change Tracking

This section identifies changes that were made to the [MS-ASPROV] protocol document between the January 2012 and April 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2 Message Syntax	Added nillable attribute to multiple elements.	N	Content updated.

8 Index