



Security Model For Windows Mobile 5.0 and Windows Mobile 6

Date February 2007
Applies to: Windows Mobile Version 5.0
Windows Mobile 6

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, ActiveSync, Authenticode, Outlook, Windows, Windows Mobile are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

Security Model	1
Naming Conventions	1
Protection Against Threats and Risks	1
Permissions.....	4
Security Configuration.....	4
Security and CAB Signing.....	6
Executables and DLL Signing	7
Other Resources	7
Security Policies for Windows Mobile 5.0 and Windows Mobile 6	9
Protecting Devices with Security Policies.....	9
Security Roles for Windows Mobile 5.0 and Windows Mobile 6	15
Additional Security Settings	19
Device Wipe.....	19
Local Wipe.....	19
Remote Wipe	20
Lock a Device.....	20
Authentication with LASS and LAP	20
Enhanced PIN Strength.....	21
Password/PIN Expiration.....	21
User PIN Reset	22
Password History	22
Certificates for Windows Mobile 5.0 and Windows Mobile 6	25
Certificates Shipped on Windows Mobile Powered Devices.....	25
Certificate Stores.....	26
Adding Certificates to Windows Mobile Powered Devices	28
Installing Certificates on a Windows Mobile 5.0-based Device	28
Installing Certificates on a Windows Mobile 6 Powered Device	30
Certificate Chains.....	31
Certificate-based Authentication	31
Managing Certificates with the CertificateEnroller Configuration Service Provider	32
Using Desktop Enrollment	32
Revoking a Certificate for a Signed Application.....	33
Security Services for Windows Mobile 5.0 and Windows Mobile 6.....	34
Cryptographic Services and FIPS Compliance in Windows Mobile 5.0 and Windows Mobile 6 ..	37

This page left intentionally blank

Security Model

Windows Mobile-powered devices employ a combination of security policies, roles, and certificates to address configuration, remote access, and application execution.

Security policies provide the flexibility to control the level of security on the device. The following list shows some of the ways you can use security policies on Windows Mobile-powered devices:

- Control which applications are allowed to run on the device and what they can do.
- Control who can access specific device settings, and their level of access.
- Control what desktop applications can do on the device (Remote API (RAPI) control through ActiveSync).

Policies configure security settings that are then enforced with the security roles and certificates:

- Security roles determine access to device resources, based on message origin and how the message is signed.
- Certificates are used to sign executables, DLLs, and CAB files that run on Windows Mobile-powered devices.

Naming Conventions

This document supports both Windows Mobile 5.0 and Windows Mobile 6.

With the introduction of Windows Mobile 6, Microsoft changed its naming conventions to better align the brand and products with the realities of today's mobile device marketplace. The historical form-factor based distinction between Windows Mobile powered Smartphone and Windows Mobile powered Pocket PC Phone is blurring dramatically, and the terminology has been changed to better reflect the evolution of the industry. The following table summarizes the changes.

Windows Mobile 5.0 and earlier	Windows Mobile 6
Windows Mobile for Pocket PC	Windows Mobile 6 Classic
Windows Mobile for Pocket PC Phone Edition	Windows Mobile 6 Professional
Windows Mobile for Smartphone	Windows Mobile 6 Standard

Protection Against Threats and Risks

The following features help protect devices against a variety of threats and risks:

Threat or Risk	Microsoft Security Features	WM 5.0	WM 5.0 with MSFP	WM 6
Access to data because of device theft or loss	Strong device password protection	X	X	X
	Device lock requires a password or PIN to access the device when it is turned on	X	X	X
	Local device wipe occurs after a specified number of incorrect login attempts		X	X
	Remote device wipe erases data and prevents unauthorized use		X	X
	Exponential back-off if incorrect passwords are entered	X	X	X
	Local storage card wipe erases data and prevents unauthorized use			X
	Remote storage card wipe erases data and prevents unauthorized use			X
	Storage card encryption prevents unauthorized use			X
	Custom Local Authentication Subsystem (LAS) and Local Authentication Plug-in (LAP) provide the infrastructure for authentication by sophisticated third-party hardware and software methods.	X	X	X
	Password policy enforcement, such as required password for synchronization		X	X
Access to data during transmission	Secure Sockets Layer (SSL) encryption of all data transmitted between the device and the corporate mail server	X	X	X
	Advanced Encryption Standard for SSL channel encryption in 128 and 256 bit cipher strengths.			X
	Encrypted data passes through a single SSL port on the firewall	X	X	X
	Cryptographic implementations certified by US Federal Information Processing Standard (FIPS) 140-2, and are protected against a variety of potential threats. Supported	X	X	X

	<p>algorithms include:</p> <ul style="list-style-type: none"> • Advanced Encryption Standard (AES) • DES and 3DES, • Secure Hash Algorithm (SHA-1), • RSA public-key encryption and decryption. 			
Unauthorized penetration into corporate network	Flexible client authentication: SSL TLS, Exchange ActiveSync, Certificate-based, RSA SecurID-protected	X	X	X
	Users can add root certificates without compromising device management security and without being a manager of the device			X
Unauthorized penetration into mobile device	Security policies control over-the-air access to device	X	X	X
	Bluetooth discovery mode can be prohibited to guard device integrity (Supported in Windows Mobile 6 Standard only)			X
Device corruption	<p>Security policies control acceptance of unsigned attachments, applications, or files</p> <ul style="list-style-type: none"> • Two-tier access for code execution control · executable runs if it is signed; permissions indicate access. (Supported by Windows Mobile-based Smartphone and Windows Mobile 6 Standard only) • One-tier access for code execution control · executable runs if it is signed. 	X	X	X
	Attachments for download can be denied or size-restricted			X
Malicious software or viruses on mobile devices	Office Mobile applications do not support macros, so viruses cannot leverage them to do damage	X	X	X
	Code execution control allows the device to be locked so that only applications signed with a trusted certificate can run	X	X	X

Permissions

Application execution is based on permissions. Windows Mobile-powered devices have a two tiered permission model, or applications can be blocked:

- Privileged
- Normal
- Blocked

Applications running at the privileged level have the highest permissions: they can call any API, write to protected areas of the registry, and have full access to system files. Few applications need to run as privileged. In fact, allowing them to run privileged allows them to change the operating system environment, and can threaten the integrity of the device.

Most applications run normal. They cannot call trusted APIs, write to protected areas of the registry, write to system files or install certificates to protected stores. They could still install a certificate to the MY store, however.

Applications do not run if blocked because they are not allowed to execute. An application could be blocked because it is not signed by an appropriate certificate, because the user blocks it after being prompted, and so forth.

Security Configuration

The security policy of a particular device determines how the device handles application signatures and permission. The first part of the security policy is known as access tiers; devices can have one-tier or two-tier access.

Security policy settings	Description
One-tier access	<p>A device with one-tier access focuses only on how an application should run based on whether the application is signed with a certificate in the device certificate store. There is no concern with permission restriction.</p> <p>Signed with a known certificate</p> <p>Signed applications execute with no further checks and run with privileged permissions on the device. They can call any API, modify any part of the file system, and modify any part of the registry.</p> <p>In a one-tier device, an application that is allowed to be run on the device has the MANAGER role.</p> <p>Unsigned or signed with an unknown certificate</p> <p>Unsigned applications or those signed with a certificate that the device does not recognize require further policy checks to determine if they can run, then security policies are checked to determine whether to prompt the user to allow them to run. If they are allowed to run, they run with the same permissions as if they were a signed application.</p>
Two-tier access	<p>A device with two-tier access restricts application start and run-time permissions.</p> <p>Signed with a known certificate</p> <p>Applications signed with a certificate that the device recognizes execute with no further checks.</p>

	<p>There is a distinction between Privileged and Normal applications:</p> <ul style="list-style-type: none"> • Applications signed with a certificate from the Privileged Execution Trust Authorities certificate store execute with privileged permissions and therefore have full access to the device. • All other applications run with normal permissions. Applications running with normal permissions can read from protected areas of the registry, and read contents of files marked with the System attribute. They cannot write to protected areas of the registry, to system files, or access files in the \Windows\System directory. <p>In two-tier device, Applications running privileged have the MANAGER role, and those running normal have the USER_AUTH role. Most applications only require normal permissions to run.</p> <p>Unsigned or signed with an unknown certificate</p> <p>Unsigned applications or those signed with a certificate that the device does not recognize require further policy checks to determine if they can run. If they are allowed to run, they will run with normal permissions.</p>
--	--

The following table shows the platform support for these security configurations:

Platform	Supports one tier	Supports two tier
Windows Mobile 5.0 powered Smartphone	Yes	Yes (default)
Windows Mobile 5.0 powered Pocket PC Phone	Yes (default)	No
Windows Mobile 5.0 powered Pocket PC	Yes (default)	No
Windows Mobile 6 Professional	Yes (default)	No
Windows Mobile 6 Classic	Yes (default)	No
Windows Mobile 6 Standard	Yes	Yes (default)

The one-tier and two-tier access model works with the next two parts of the security policy:

- Whether unsigned applications can execute
- Whether the user should be prompted before the unsigned application executes.

Together these settings create the following common security levels:

Security level	Description
Security off	<p>Both signed and unsigned applications are allowed to run with no further security checks and without prompting the user.</p> <p>(This is a one-tiered policy.)</p> <p>Any application can call any API, and modify any part of the registry and file system.</p> <p>This policy may be used during the testing phase of a device, but it is not a safe policy to have on an actual device. A device with this</p>

	<p>policy would be vulnerable to malicious applications and allow unrestricted access to the device.</p>
One-tier prompt	<p>This policy allows applications signed with a certificate recognized by the device to execute with no user prompt, since the application's certificate matched a device certificate.</p> <p>The device prompts the user before allowing unsigned or incorrectly signed applications to run.</p> <p>Once a signed or unsigned application is running, it has full permissions on the device.</p>
Two-tier prompt	<p>This policy allows signed application to execute. The device prompts the user before executing unsigned applications.</p> <p>Once a signed application is executing, the application permissions are determined by the certificate:</p> <ul style="list-style-type: none"> • Applications signed with a certificate in the Privileged Execution Trust Authorities store have unrestricted access to the device. • Applications signed with a certificate from the Unprivileged Execution Trust Authorities store run with normal permissions. They can read from protected areas of the registry, and read contents of files marked with the System attribute. They cannot write to protected areas of the registry, to system files, or access files in the \Windows\System directory. <p>If a user allows an unsigned application to execute, it executes with normal permissions.</p>
Mobile2Market locked	<p>Applications that are signed can execute; Unsigned applications are not allowed to execute.</p> <p>Once the application is running, the permissions are determined by whether the application is signed with a certificate from the Privileged Execution Trust Authorities store or the Unprivileged Execution Trust Authorities store.</p> <p>Mobile2Market is the Microsoft certification and marketing program for mobile applications for independent software and hardware vendors. Mobile2Market partners provide certificate authority and digital signature services for Windows Mobile. Once certified, applications can be marketed through Mobile2Market's Mobile Application catalog.</p>
Locked	<p>Market2Market certificates have been removed from the device, but OEM, Mobile Operator, or Enterprise certificates are present.</p>

Security and CAB Signing

Cabinet (.cab) files are used to package applications or updates, such as new DLL files, for delivery. Depending on security policy settings, you may need to sign .cab files to install the contents. You can use Microsoft Authenticode tools to sign .cab files.

Note Files signed using the Authenticode tool can be later revoked.

If the policy settings require signed .cab files, the one creating the cab must confirm the following:

- The revocation list does not include the certificate hashes. A certificate hash is a digest of the certificate data.
- The revocation list does not include the .cab file hashes.
- The certificate chain maps to a root certificate in the Software Publisher Certificate (SPC) store.

The .cab file is installed with the role mask associated with the root certificate. If the store does not contain a matching root certificate, the .cab file is treated as an unsigned .cab file. The Unsigned CABS policy determines whether the file can be installed and with which role mask an accepted file is installed

Executables and DLL Signing

Executables and DLLs are signed and validated against certificates in the device Privileged or Unprivileged certificate stores.

The following list shows the behavior based on permissions:

- Privileged executables cannot access DLLs that have Normal permissions.
- Normal executables can load Privileged DLLs, but the DLL will run at Normal level. This is because the privilege is assigned to processes rather than to modules.

Other Resources

Security for Windows Mobile Messaging in the Enterprise

<http://www.microsoft.com/technet/solutionaccelerators/mobile/default.mspx>

This page left intentionally blank

Security Policies for Windows Mobile 5.0 and Windows Mobile 6

Security Policy settings on Windows Mobile powered devices define levels of security. For example, policies determine whether a device can be configured over the air (OTA), and whether to accept unsigned messages, applications, or files. Security policy settings provide flexible control over the level of software security. These policies are defined globally and enforced locally in their respective components at critical points across the device architecture.

Security roles allow or restrict access to device resources. Roles define who can change each policy. The Manager role allows complete control over the device. For a list of the roles, see [Security Roles for Windows Mobile 5.0 and Windows Mobile 6](#).

By default, only someone with Manager role permissions on the device can change most of the security policies. Using Exchange ActiveSync, network administrators can change a few policies. Additionally, if the OEM has given a mobile operator or network administrator Manager role permissions, they can change all security policies on the device by provisioning it.

You can manage a device by provisioning it. Provisioning is updating the device after manufacture; this may or may not include bootstrapping a device. Provisioning a device involves creating a provisioning XML file that contains configuration information, and then sending the file to the device. Configuration Service Providers then configure the device based on the contents of the XML file.

Protecting Devices with Security Policies

The following table shows how you can use security policies to protect devices; the policy ID is shown in parentheses. Security roles define who can change each policy; the default role is listed.

Protection	Microsoft Security Policy
Block unauthorized penetration into device	<ul style="list-style-type: none"> <li data-bbox="634 1104 1377 1276">• Allow or deny permission for applications stored on a Multimedia Card (MMC) to run automatically when inserted into the device (2) Default Setting: allow Default Role: Manager <li data-bbox="634 1287 1325 1402">• Allow or deny unsigned .cab files to be installed (4101) Default Setting: allow for SECROLE_USER_AUTH Default Role: Manager <li data-bbox="634 1413 1263 1528">• Allow or deny unsigned applications to run (4102) Default Setting: allow Default Role: Manager <li data-bbox="634 1539 1373 1682">• Allow or deny unsigned theme files to be installed, or allow only unsigned theme files with a specific role mask (4103) Default Setting: allow for SECROLE_USER_UNAUTH Default Role: Manager <li data-bbox="634 1692 1349 1873">• Allow or deny Service Loading (SL) messages as a role mask. An SL message automatically downloads the new service, update, or provisioning file. (4108) Default Setting: allow for SECROLE_PPG_TRUSTED Default Role: Manager

	<ul style="list-style-type: none">• Allow or deny Service Indication (SI) messages. An SI message is sent to the connected device to notify users of new services, service updates, and provisioning services. (4109) Default Setting: allow for SECROLE_PPG_AUTH SECROLE_PPG_TRUSTED Default Role: Manager• Allow or deny unsigned WAP messages processed by a specific role mask. (4110) Default Setting: allow for SECROLE_USER_UNAUTH Default Role: Manager• Specify which over-the-air (OTA) provisioning messages to accept based on roles assigned to the messages. (4111) Default Setting: Allow for SECROLE_OPERATOR_TPS SECROLE_PPG_TRUSTED SECROLE_PPG_AUTH SECROLE_TRUSTED_PPG SECROLE_USER_AUTH SECROLE_OPERATOR Default Role: Manager• Allow or deny the routing of Wireless Session Protocol (WSP) notifications from the WAP stack. (4113) Default Setting: allow Default Role: Manager• Specify whether to prompt a user to accept or reject unsigned .cab, theme, .dll, and .exe files. (4122) Default Setting: prompt user Default Role: Manager• Specify whether to prompt the user to confirm changes in device settings when an over-the-air (OTA) OMA Client Provisioning message is signed with only a network personal identification number (PIN). (4132) Default Setting: do not prompt user Default Role: Manager <p>Applies to Windows Mobile 5.0 and Windows Mobile 5.0 with MSFP:</p> <ul style="list-style-type: none">• Specify roles on which to base acceptance of a WAP signed OMA Client Provisioning message. (4107) In Windows Mobile 6, this policy is deprecated; use 4141, 4142, and 4143 instead. Default Setting: SECROLE_PPG_AUTH SECROLE_PPG_TRUSTED SECROLE_OPERATOR_TPS Default Role: Manager <p>Applies to Windows Mobile 5.0 with MSFP only</p> <ul style="list-style-type: none">• Specify that the user must always authenticate on the device to unlock it, or allow the user to enter a PIN on the desktop. (4133) In Windows Mobile 6, this policy is
--	---

	<p>deprecated; use 4146 instead. Default Setting: allow PIN on desktop Default Role: Manager, Enterprise</p> <p>Applies to Windows Mobile 6:</p> <ul style="list-style-type: none"> • Allow or deny the user permission to change mobile encryption settings for removable storage media. (4134) Default Setting: allow Default Role: Manager, Enterprise • Allow or deny other devices permission to search Bluetooth-enabled devices (4135) Default Setting: Bluetooth device can be set to discoverable Default Role: Manager • Allow or deny Outlook Mobile permission to get documents on a corporate Sharepoint® or UNC through ActiveSync (4145) Default Setting: deny Default Role: Manager • Specify whether or not the user must authenticate on the device when connected if device lock is active (4146) Default Setting: can authenticate through shared secret on desktop Default Role: Manager, Enterprise • Allow or deny OMA Client Provisioning network PIN message. (4141) Default Setting: allow for SECROLE_PPG_AUTH SECROLE_PPG_TRUSTED SECROLE_OPERATOR_TPS Default Role: Manager • Allow or deny an OMA Client Provisioning user PIN or user MAC signed message. (4142) Default Setting: allow for SECROLE_PPG_AUTH SECROLE_PPG_TRUSTED SECROLE_OPERATOR_TPS Default Role: Manager • Allow or deny an OMA Client Provisioning user network PIN signed message. (4143) Default Setting: Allow for SECROLE_PPG_AUTH SECROLE_PPG_TRUSTED SECROLE_OPERATOR_TPS. Default Role: Manager
<p>Protect against application corruption</p>	<ul style="list-style-type: none"> • Allow or deny access of remote applications that are using Remote API (RAPI) to implement ActiveSync operations, or restrict RAPI ActiveSync access to User Authenticated role. (4097) <p style="text-align: center;">Note RAPI being unrestricted means that the user has</p>

	<p>Manager permissions on the device. Default Setting: allow for SECROLE_USER_AUTH Default Role: Manager</p>
<p>Protect sensitive data during transmission</p>	<p>Many of the policies that protect data during transmission are used in Secure Multipurpose Internet Mail Extensions (S/MIME), which allows you to encrypt or digitally sign e-mail messages. S/MIME encryption and/or digital signing are available when sending e-mail via Outlook 2003 or Outlook Web Access (OWA).</p> <p>In Windows Mobile 6, the device has full support for S/MIME. However, for a Windows Mobile 6 device to view and send S/MIME messages in a supported way, the device must be synchronizing against an Exchange 2003 SP2 server.</p> <p>Applies to Windows Mobile 5.0 with MSFP only</p> <ul style="list-style-type: none"> • Specify whether the Inbox application will sign all messages and, if so, the algorithm used for signing. This policy is used in S/MIME. (4125) In Windows Mobile 6, this policy is deprecated; use 4137 and 4139 instead. Default Setting: do not sign Default Role: Manager • Specify whether the Inbox application will encrypt all sent messages and, if so, the algorithm to use for encryption. This policy is used in S/MIME. (4126) In Windows Mobile 6, this policy is deprecated; use 4138 and 4140 instead. Default Setting: do not encrypt Default Role: Manager <p>Applies to Windows Mobile 5.0 with MSFP and later</p> <ul style="list-style-type: none"> • Allow or deny software certificates to be used to sign outgoing messages. This policy is used in S/MIME. (4127) Default Setting: allow Default Role: Manager <p>Applies to Windows Mobile 6:</p> <ul style="list-style-type: none"> • Allow or deny Inbox application to negotiate the encryption algorithm when the specified encryption algorithm is not supported. (4144) Default Setting: do not negotiate Default Role: Manager • Allow or deny HTML messages. (4136) Default Setting: allow Default Role: Manager • Require encryption of Inbox S/MIME messages. (4138) Default Setting: encryption is optional Default Role: Manager • Make signing of Inbox S/MIME messages required or

	<p>optional. (4137) Default Setting: optional Default Role: Manager</p> <ul style="list-style-type: none"> • Specifies the algorithm used to sign a message. This policy is used in S/MIME.(4139) Default Setting: sign with the default algorithm Default Role: Manager • Specify the algorithm used to encrypt a message. This policy is used in S/MIME.(4140) Default Setting: encrypt with the default algorithm Default Role: Manager
<p>Protect sensitive data in case of device theft or loss</p>	<ul style="list-style-type: none"> • Specify the maximum number of times the user is allowed to try to authenticate a Wireless Application Protocol (WAP) PIN-signed message. (4105) Default Setting: 3 Default Role: Manager <p>Applies to Windows Mobile 5.0 with MSFP and later</p> <ul style="list-style-type: none"> • Specify whether a password must be configured on the device. (4131). Default Setting: password required Default Role: Manager, Enterprise
<p>Specify security level</p>	<ul style="list-style-type: none"> • Grant User Authenticated system administrative privileges to other security roles specified with role mask. (4120) Default Setting: SECROLE_USER_AUTH Default Role: User Authenticated • Specify which DRM rights messages are accepted by the DRM engine based on the role assigned to the message. (4129) Default Setting: accept from SECROLE_PPG_AUTH SECROLE_PPG_TRUSTED Default Role: Manager • Grant Manager system administrative privileges to other security roles. (4119) <p style="margin-left: 40px;">Note Any role added to this policy becomes a Manager of the device. For example, if User Authenticated role is added, the user becomes a manager to the device.</p> Default Setting: SECROLE_OPERATOR_TPS for Windows Mobile 6 Professional; SECROLE_USER_AUTH for Windows Mobile 6 Classic; OPERATOR_TPS for Windows Mobile 6 Standard Default Role: Manager • Specify which application access model is implemented on the device (one-tier or two-tier) (4123) Default Setting: two tier access for Windows Mobile 6

	<p>Professional; one tier for Windows Mobile Standard. Default Role: Manager</p> <ul style="list-style-type: none">• Specify whether mobile operators can be assigned the Trusted Provisioning Server (TPS) role. (4104) Default Setting: TPS role disabled Default Role: Manager• Specify the permissions required to create, modify, or delete a trusted proxy. (4121) Default Setting: allow for SECROLE_OPERATOR SECROLE_OPERATOR_TPS SECROLE_MANAGER Default Role: Manager• Allow operator to override https to use http or wsp to use wsp. (4124) Default Setting: use http or wsp Default Role: Manager
--	--

Security Roles for Windows Mobile 5.0 and Windows Mobile 6

Security roles allow or restrict access to device resources. For example, roles are used to determine whether a remote message is accepted, and if it is, what level of access it is allowed. Roles are also used to provide access to each Configuration Service Provider. Configuration Service Providers manage configuring the device during the provisioning process.

Note OEMs and Mobile Operators control access to Configuration Service Providers. The following table shows Windows Mobile 5.0 and Windows Mobile 6 security roles.

Security Roles	Description
None (SECROLE_NONE)	No role assigned.
Manager (SECROLE_MANAGER)	Setting can be changed by the manager or administrator. This role allows unrestricted access to system resources.
Enterprise (SECROLE_ENTERPRISE)	<p>Applies to Windows Mobile 5.0 with MSFP and later</p> <p>Exchange Administrator role. The Enterprise role allows IT administrators to manage specific device settings, such as wiping a device, setting password requirements, and managing certificates.</p> <p>Example of use: Using this role with the Message Authentication Retry Number policy allows the Enterprise IT Professional to change the policy setting.</p>
Operator (SECROLE_OPERATOR)	<p>Setting can be changed by a Wireless Application Protocol (WAP) Trusted Provisioning Server (TPS).</p> <p>Example of use: Using this role with the Auto Run Policy allows the Mobile Operator to change the policy; the Operator would be able to allow or restrict applications stored on a Multimedia Card (MMC) to automatically run when inserted into the device.</p>
Authenticated User (SECROLE_USER_AUTH)	<p>Setting can be changed by an authenticated user. This role can be assigned to the device owner.</p> <p>Permissions are determined by the settings to which the user requires access. Typically, this setting is assigned to:</p> <ul style="list-style-type: none"> • User PIN-signed WAP push messages. • Messages received through the Remote API (RAPI) by default. <p>The user can query device information, manage files and directories, and change settings such as the home screen and sounds.</p> <p>Applies to Windows Mobile 6</p> <p>The owner can manage user certificates and designated certificate stores.</p> <p>Example of use: Use this role with a security policy to allow the user to configure the setting associated with the policy.</p>

<p>Unauthenticated User (SECROLE_USER_UNAUTH)</p>	<p>Setting can be changed by anyone.</p> <p>Assigned to unsigned WAP push messages. This role provides permissions to install Home/Today screen or ring tones.</p> <p>Example of use: Use this role with the Unsigned Theme security policy to allow users to install unsigned themes on their device; Themes are used for processing homescreens.</p>
<p>Trusted Provisioning Server (SECROLE_OPERATOR_TPS)</p>	<p>OMA Client Provisioning messages that come from a WAP Push Initiator that is authenticated by a trusted Push Proxy Gateway, and where the Uniform Resource Identifier (URI) of the Push Initiator corresponds to the URI of the Trusted Provisioning Server (TPS) on the device.</p> <p>Example of use: Use this role to grant system administrative privileges to the Mobile Operator's trusted provisioning server (TPS).</p>
<p>Known Push Proxy Gateway (SECROLE_KNOWN_PPG)</p>	<p>Messages assigned this role indicate that the device knows the address to the Push Proxy Gateway used in provisioning.</p> <p>Example of use: Using this role for the Service Indication Message Policy means that the device only accept SI message from known Push Proxy Gateway</p>
<p>Device Trusted Push Proxy Gateway (SECROLE_PPG_TRUSTED)</p>	<p>Messages assigned this role indicate that the content sent by the Push Initiator is trusted by the Push Proxy Gateway. This role implies that the device trusts the Push Proxy.</p> <p>Example of use: Using this role along with SECROLE_PPG_AUTH and the DRM security policy means that unauthenticated messages are not accepted by the device. Content from the push router is filtered out based on the trust of the message origin.</p>
<p>Push Initiator Authenticated (SECROLE_PPG_AUTH)</p>	<p>Messages assigned this role indicate that the Push Proxy Gateway is known and trusted by the device.</p> <p>Since WAP secure push is not supported, the Push Proxy Gateway is not currently authenticated. The address of the Push Proxy Gateway is compared with the trusted Push Proxy Gateway address stored on the device.</p> <p>Example of use: Using this role along with SECROLE_PPG_TRUSTED and the DRM security policy means that unauthenticated messages are not accepted by the device. Content from the push router is filtered out based on the trust of the message origin.</p>
<p>Trusted Push Proxy Gateway (SECROLE_TRUSTED_PPG)</p>	<p>Messages assigned this role indicate that the content sent by the Push Initiator is trusted by the Push Proxy Gateway. This role implies that the device trusts the Push Proxy Gateway.</p> <p>Example of use: Using this role for the Service Loading Message Policy means that the device only accept SL message from trusted Push Proxy Gateway. An SL message downloads new services or provisioning XML to the Windows Mobile powered device.</p>

<p>Any Push Message (SECROLE_ANY_PUSH_SOURCE).</p>	<p>Applies to Windows Mobile 6</p> <p>Messages received by the push router will be Example of use: Adding this role to OMA Client Provisioning Network PIN Policy means that the OMA Network PIN signed message will be accepted.</p>
--	--

This page left intentionally blank

Additional Security Settings

The main elements of the Windows Mobile security model are security policies, roles, and certificates. There are, however, additional methods of protecting the device, such as by registry settings, or by the Enterprise administrator using Exchange Server ActiveSync.

Device Wipe

When a mobile device is lost or stolen, the potential security risk can be significant. Mobile devices often contain sensitive business data, including personally identifiable information of employees and customers, sensitive e-mail messages, and other items that could have a negative impact on the business. Exchange ActiveSync addresses this risk by providing two levels of device wipe capability.

Wiping the device locally or remotely has the effect of performing a factory or hard+reset; all programs, data, and user-specific settings are removed from the device. The Windows Mobile device wipe implementation wipes all data, settings, and private key material on the device by overwriting the device memory with a fixed bit pattern, greatly increasing the difficulty of recovering data from a wiped device.

Note Device wipe in Windows Mobile 6 powered devices includes wiping the removable storage card.

Local device wipes are triggered on a device with device lock enforced if a user incorrectly enters a PIN more than a specified number of times (the policy default is 8 times, but the administrator can adjust this value). After every two missed attempts, the device displays a confirmation prompt that requires the user to type a confirmation string (usually %A1B2C3+) to continue. This prevents the device from being wiped by accidental key presses. Once the PIN retry limit is reached, the device immediately wipes itself, erasing all local data.

Remote wipes occur when the administrator issues an explicit wipe command through the Exchange ActiveSync management interface. With OWA 2007 and Exchange Server 2007, the device user can also initiate a wipe command if they've lost their device. Remote wipe operations are separate from local wipes, and a device can be wiped remotely even if Exchange ActiveSync security policies are not in force. The wipe command is pushed as an out-of-band command, so that the device receives it on its next synchronization. The device sends an acknowledgement message when it receives the wipe command, alerting the administrator that the wipe has occurred. The device user cannot opt out of the remote wipe.

Local Wipe

Local device wipe is accomplished by using both the Password Required Policy (4131) and the following registry key:

Registry\HKLM\Comm\Security\Policy\LASSD\DeviceWipeTreshold

This setting's value is the number of incorrect password attempts to allow before the device's memory is securely erased. The value can be 1 through 4294967295 (0xFFFFFFFF).

This registry key does not exist by default. If it does not exist, is set to zero (0), or is set to 4294967295 (0xFFFFFFFF), the local wipe feature is turned off.

The Manager and Enterprise role can change this setting. This setting corresponds with a setting available on the Device Security Settings dialog box in Exchange Server 2003 SP2.

Note Microsoft recommends that you also enforce authentication from the device by using one of the following policies:

- For Windows Mobile 5.0 and Windows Mobile 5.0 with MSFP, use Desktop Unlock Policy (4133)

- For Windows Mobile 6, use Desktop Quick Connect Authentication Policy (4146)

Remote Wipe

The Exchange administrator can use the Exchange Server Configuration Tools directly to wipe the device even when Password Required or DeviceWipeThreshold is not enforced.

Note The Hint that appears for users to remind them of their password, is hard coded to appear after five incorrect password attempts. Exchange Administrators must take this into consideration, and configure Remote Wipe to occur after users see their Hint.

Lock a Device

Locking a device after inactivity is the interaction of the following features:

- Password and PIN Expiration
- Sequences and Patterns in Passwords and PINs
- Password History
- Enhanced PIN Strength

These settings are enforced through Local Authentication Subsystem (LASS) and Local Authentication Plug-ins (LAP).

The IT administrator configures policies and device requirements in Exchange System Manager interface in Exchange Server 2003 SP2 or Exchange Server 2007 ActiveSync Mailbox Policy wizard. They configure the following information:

- Whether to require the user to automatically lock the device after a period of inactivity.
- The maximum amount of idle time before requiring the device to lock.
- The minimum password strength and length required

This information is saved in a protected portion of the registry.

The user configures their device with settings that meet the minimum requirements set by the IT administrator, including configuring their password or PIN, and setting the length of inactivity time before the device locks.

If the IT administrator places a recovery pass code in Outlook Web Access (OWA), users can create a new device password or PIN if they forget the one that they chose.

Authentication with LASS and LAP

Local Authentication Subsystem (LASS) allows flexible integration of Local Authentication Plug-ins (LAPs).

LASS provides the infrastructure for authentication by sophisticated third-party hardware and software methods, including biometrics, Smartcard use, a hardware button combination, or user signature. LASS can also be used to specify event-based policies to authenticate users. For example, device lock can be triggered programmatically, not just when a device is turned on.

A LAP is an authentication mechanism that plugs into LASS. Windows Mobile 5.0 and later contains a built-in password LAP. OEMs and ISVs can build custom pluggable authentication modules.

The Microsoft LAP provides two types of password enforcement that can be configured with policies on the server: a minimum password length, and either a strong alphanumeric password or simple PIN.

Note If a third-party solution is added to the Device Unlock behavior, the behavior of the device may change for the end user and it may be a less secure solution. If possible,

OEMs and Mobile Operators should ask third-party vendors and Enterprise Administrators whether they prefer authentication on the desktop or on the device.

Applies to Windows Mobile 6:

The following list describes the additional LAP functionality in Windows Mobile 6:

- **Enhanced PIN Strength.** Enhanced PIN Strength in Windows Mobile 6 prevents users from choosing a PIN that contains a simple pattern or has too few digits.
- **Password/PIN Expiration.** Password/PIN expiration permits setting the expiration time of a password or PIN on a device using the Microsoft Default LAP.
- **User PIN Reset.** User password/PIN on a device using the Microsoft Default LAP can be reset using an Authentication Reset Component (ARC).
- **Password History.** Password History ensures that users choose unique passwords by comparing the new password against a specified number of previous passwords.

Enhanced PIN Strength

The Microsoft Default LAP can be configured to prevent users from choosing a PIN that contains a simple pattern or has too few digits. This feature requires Microsoft Exchange Server Version 12.0.

The feature will:

- Enable a policy that requires end users to choose a PIN that does not contain a repeating sequence, such as 1111.
- Enable a policy that requires end users to choose a PIN that does not contain a sequence with a predictable difference between values, such as 1234 or 1357.
- Provide a mechanism for IT administrators to configure policies via a third-party device-management solution.

When administrators enable this policy, users are prevented from specifying a PIN with a uniform offset between successive digits. For example, when this policy is enabled users cannot set a PIN to a sequence like '1111', '1234', or '1357'.

HKEY_LOCAL_MACHINE\Comm\Security\LASSD\LAP\lap_pw\AllowSimplePIN

The following list shows the possible values:

- 0 indicates that the policy defined in the local authentication plug-in is applied.
- 1 indicates that simple PINs are allowed.

If the device has an existing device lock PIN when this policy is distributed, the user will have to enroll again because it is not possible to determine if an existing PIN value conforms to a policy or not. The PIN value is stored as a hash so there is no way to determine if an existing PIN satisfies the policy.

Password/PIN Expiration

Applies to Windows Mobile 6:

Password/PIN expiration permits setting the expiration time of a password or PIN on a device using the Microsoft Default LAP. This feature requires Microsoft Exchange Server Version 12.0.

The feature will:

- Provide a policy that requires end users to choose a new password or PIN after a configured time period (in seconds).
- Provide a mechanism for IT administrators to configure policies via a third-party device-management solution.

The Microsoft default LAP allows administrators to enforce a policy of how often a user must choose a new password. The password expiration feature is dependent on the phone clock. Once the expiration period is reached the user is prompted to change their password. The new password must meet the other requirements such as password or PIN strength and password history

The password or PIN expiration information is stored in a protected area of the registry.

User PIN Reset

Applies to Windows Mobile 6:

User password/PIN on a device using the Microsoft Default LAP can be reset using an Authentication Reset Component (ARC). Unlike the other features, the use of the ARC with a custom LAP is supported. The ARC is a pluggable component and an OEM may create an ARC for use with a custom LAP or the default LAP. This feature requires Microsoft Exchange Server Version 12.0.

The feature will:

- Provide the ability for the end user to request a reset from their administrator or by using Outlook Web Access.
- Ensure that devices lock reliably.
- Support infrastructures that use certificate authentication or rely on credentials to authenticate a user to the system.
- Support OEM customization of the LAP.
- Support OEM replacement of the ARC.

During Authentication reset, the Reset Password option appears on the password screen menu. This functionality is enabled through the following registry key:

HKEY_LOCAL_MACHINE\Comm\Policy\LASSD\AuthReset\AuthenticationReset

A value of 0 indicates that Authentication Reset is disabled; a value of 1 indicates that it is enabled.

The recovery PIN is an important element of the reset process. The recovery PIN is a 16-character alphanumeric value created during setup without user interaction. In fact, the user is not aware that the recovery PIN is created and transmitted to the Exchange Server. When a user runs setup the first time, the recovery PIN is created on the device and transmitted to the Exchange server where it is stored. The recovery PIN is used to encrypt the Master Key.

The following list describes the process for a user PIN reset:

1. The user creates a new PIN to unlock the device.
When creating the new PIN, any PIN history and strength policies are applied. The device lock policies are applied to the new password before the user is allowed to continue with the User PIN Reset process.
2. The user obtains the recovery PIN through Outlook Web Access or by calling a Helpdesk.
3. The user enters the recovery PIN and then enters the new PIN created in the first step of this process to unlock the device.

The recovery PIN is considered compromised so it is discarded following the User PIN Reset process. A new recovery PIN is created on the device and transmitted to the Exchange Server.

Password History

Applies to Windows Mobile 6:

Password History uses the Microsoft Default LAP to maintain password history and securely store passwords on the device to prevent reuse of a password. This feature requires Microsoft Exchange Server Version 12.0.

The feature will:

- Enable a policy that requires end users to choose a new password or PIN that is different from a previous password.
- Provide data about the number of stored passwords to the end user if the new password matches a previous password.
- Provide a mechanism for IT administrators to configure policies via a third-party device-management solution.

The number of previous passwords to check is contained in the following registry setting:

HKEY_LOCAL_MACHINE\Comm\Security\LASSD\LAP\lap_pw\NumberOfPasswords

The value is the number of passwords stored for historical comparison.

Users often reuse passwords when creating a new password, in part because it is difficult to invent a memorable password. Matching is exact so only a password that matches exactly (including case) will be rejected. The implementation will store the number of previous passwords as salted hashes encrypted using DPAPI.

The Exchange Server administrator can set the number of saved passwords in Exchange Server Version 12.0. Passwords are not stored until the device receives a policy and the policy is only activated when a user attempts to change the device lock password.

This page left intentionally blank

Certificates for Windows Mobile 5.0 and Windows Mobile 6

Digital certificates play a critical role in device security and network authentication. Certificates are electronic credentials that bind the identity of the certificate owner or the device to a public and private pair of electronic keys used to encrypt and digitally sign information. Signed digital certificates assure that the keys actually belong to the specified application, device, organization, or person and that they can be trusted.

Digital certificates are used on Windows Mobile devices in two essential roles:

- In code signing, determining whether an application can be run on the device and if so, the permissions (privileged or normal) with which it will run.
- In authentication, presenting trusted credentials for connecting to a corporate e-mail server or network or verifying the identity of a remote server.

For example, in order for an application to be installed and run on the device, the application must present a digital certificate that proves it was accepted and signed by a trusted source, such as the Mobile2Market program. In an authentication example, before an SSL connection can be established with the network server, the mobile device must present a certificate from its root store that is recognized and accepted by the server.

Mobile2Market is the Microsoft certification and marketing program for mobile applications for independent software and hardware vendors. This program, in conjunction with privileged certificate authorities, allows application developers to distribute their applications across the vast majority of Windows Mobile-powered devices while working with a single certificate authority and maintaining just one signed version of their application.

Certificates Shipped on Windows Mobile Powered Devices

By default, Windows Mobile-powered devices are shipped with a variety of certificates:

- Trusted root certificates from major certificate vendors that can be used for authentication purposes.
- Mobile2Market and other trusted certificates that designate applications that are signed for use on Windows Mobile devices.
- Additional certificates that may be added by the OEM or network carrier.

The following table lists the certificates shipped with Windows Mobile 5.0-based devices at this printing.

Vendor	Certificate name
Cybertrust	GlobalSign Root CA
Cybertrust	GTE CyberTrust Global Root
Cybertrust	GTE CyberTrust Root
Verisign	Class 2 Public Primary Certification Authority
Verisign	Thawte Premium Server CA
Verisign	Thawte Server CA
Verisign	Secure Server Certification Authority
Verisign	Class 3 Public Primary Certification Authority
Entrust	Entrust.net Certification Authority (2048)
Entrust	Entrust.net Secure Server Certification Authority

Geotrust	Equifax Secure Certificate Authority
Godaddy	http://www.valicert.com/

The following table lists the certificates shipped with Windows Mobile 6 powered devices at this printing.

Vendor	Certificate name
Comodo	AAA Certificate Services
Comodo	AddTrust External CA Root
Cybertrust	Baltimore CyberTrust Root
Cybertrust	GlobalSign Root CA
Cybertrust	GTE CyberTrust Global Root
Verisign	Class 2 Public Primary Certification Authority
Verisign	Thawte Premium Server CA
Verisign	Thawte Server CA
Verisign	Secure Server Certification Authority
Verisign	Class 3 Public Primary Certification Authority
Entrust	Entrust.net Certification Authority (2048)
Entrust	Entrust.net Secure Server Certification Authority
Geotrust	Equifax Secure Certificate Authority
Geotrust	GeoTrust Global CA
Godaddy	Go Daddy Class 2 Certification Authority
Godaddy	http://www.valicert.com/
Godaddy	Starfield Class 2 Certification Authority

Certificate Stores

The certificates in a Windows Mobile device are located in the certificate store in the registry.

The certificate Root and Certificate Authentication (CA) stores on Windows Mobile 5.0 devices are locked to everyone except those with Manager role permissions to ensure the integrity of the digital certificates.

In Windows Mobile 6, the certificate stores have been expanded with separate User Root and CA stores to allow device users with the less-powerful authenticated user permissions to add or to enroll for trusted digital certificates. The system Root and CA stores remain locked without Manager or Enterprise role permissions.

The following table shows the certificate stores and their uses and permissions.

Certificate store	Physical Store	Description
Privileged Execution Trust Authorities	HKLM	Contains trusted certificates. Applications signed with a certificate from this store will run

		with privileged trust level (Trusted).
Unprivileged Execution Trust Authorities	HKLM	Contains normal certificates. On a one-tier device, an application signed with a certificate in this store will run with privileged trust level (Privileged). On a two-tier device, applications signed with a certificate from this store will run with normal level (Normal).
SPC	HKLM	Contains Software Publishing Certificates (SPC) used for signing .cab or .cpf files and assigning the correct role mask to the file installation.
Root (system)	HKLM	Contains root certificates, which can be certificates signed by Microsoft, the OEM, or the Mobile Operator. These certificates are used for SSL server authentication. These cannot be changed without Manager role permissions. Users with Manager role can add certificates in this store. OMA-DM transport client only checks this store for root certificates when establishing an SSL connection.
Root (user)	HKCU	Applies to Windows Mobile 6: Contains root, or self-signed, certificates that can be installed by someone with Authenticated user role.
CA (system)	HKLM	Contains certificates and information from intermediary certification authorities. They are used for building certificate chains. In Windows Mobile 5.0, users with Manager role can add certificates in this store. OMA-DM transport client only checks this store for intermediate certificates when establishing an SSL connection. Applies to Windows Mobile 6: Certificates are added to this store by Microsoft, the OEM, or the Mobile Operator.
CA (user)	HKCU	Applies to Windows Mobile 6: Contains certificates, including those from intermediary certification authorities, which can be installed by the device user with Authenticated User role. They are used for building certificate chains.
MY	HKCU	Contains end-user personal certificates used for certificate authentication or S/MIME.

Adding Certificates to Windows Mobile Powered Devices

Because digital certificates are a key component of the Windows Mobile security model, there are restrictions on the permissions and requirements for adding and managing certificates.

The processes and tools for adding and managing certificates are different for Windows Mobile 5.0 and Windows Mobile 6.

Installing Certificates on a Windows Mobile 5.0-based Device

Network managers and device users can place additional certificates on Windows Mobile 5.0 devices if they have either Manager permission on the device or have the ability to run trusted code.

OEMs and mobile operators determine the security policies shipped on their devices. Confer with your device vendor or mobile operator to ensure that the devices you intend to purchase will either work with the certificates you currently have deployed, that you can add the necessary certificates, or that you can replace your certificates in a cost-effective fashion.

If you wish to install root certificates for certificate-based authentication, you can use the tool for deploying Exchange ActiveSync certificate-based authentication; it can be downloaded from the following Microsoft Download center Web site.

For more information, see the Microsoft Knowledge Base article, How to install root certificates on a Windows Mobile-based device available at the following Microsoft Web site:

<http://support.microsoft.com/kb/915840>.

Note Windows Mobile 5.0 does not support the use of wildcard certificates for device-to-server authentication. This restriction applies to all communications, including Exchange ActiveSync.

Note In Windows Mobile 5.0, users cannot disable SSL checking for Exchange ActiveSync (EAS) by running the certchk tool because it was being used to circumvent SSL security.

Installing a Root Certificate on a Windows Mobile 5.0-based device

- Convert the root certificate (.cer) to a Base-64 Encoded x.509 certificate.
- Create a provisioning file to install the certificate on the device.
- Deliver the certificate to the device.

Converting the Root Certificate

You must first find the desired root certificate, import it, and then export it so that it can be passed to the mobile device.

To convert the certificate

1. Open Microsoft Management Console, and then expand the Certificates . Current User node. If the node does not appear, you need to add the Certificates snap-in.
2. Choose Personal, and then double-click Certificates.
3. On the Action menu, point to All Tasks and then choose Import.
4. Follow the onscreen instructions to import the root certificate, and do the following:
 - a. Import the root certificate file (.cer) as a x.509 certificate (*.cer;*.crt) file.
 - b. Add the imported root certificate to the Personal certificate store displayed in Microsoft Management Console.
5. Choose the imported root certificate.

6. On the Action menu, point to All Tasks and then choose Export.
7. Follow the onscreen instructions to export the root certificate, and then do the following:
 - a. Export the root certificate as a base-64 encoded X.509(.CER) file.
 - b. Save the exported certificate file in the same folder as the imported certificate file.

Creating a Provisioning File

This XML file includes the instructions and specific locations for adding the certificate to the desired certificate store on the device.

Note Your provisioning XML must not contain Byte Order Markers (BOM). Use a text editor that does not insert BOMs when saving files in UTF-8 format.

Create a provisioning file\

1. Add the following XML code to a document:


```
c. <wap-provisioningdoc>
d.   <characteristic type="CertificateStore">
e.     <characteristic type="ROOT">
f.       <characteristic type="CERTHASH">
g.         <parm name="EncodedCertificate" value="BASE64ENCODEDCERT"/>
h.       </characteristic>
i.     </characteristic>
j.   </characteristic>
k. </wap-provisioningdoc>
```
2. Replace STORELOCATION with ROOT.
3. In Windows Explorer, double-click the exported root certificate.
4. Choose the Details tab.
5. Choose Thumbprint in the list box, select the text, and then press CTRL+C.
6. In the XML code, to add the root certificate thumbprint to the provisioning XML, replace CERTHASH with the copied text.
7. Delete the spaces in the thumbprint text.
8. Open the exported root certificate using a text editor.
9. Delete BEGIN CERTIFICATE and END CERTIFICATE, and then remove line breaks from the remaining text. This text is the encoded contents of the root certificate.
10. Select the text, and then press CTRL+C.
11. In the XML code, to add the root certificate to the provisioning XML, replace BASE64ENCODEDCERT with the copied text. The completed provisioning XML document will appear as shown in the following example.


```
l. <wap-provisioningdoc>
m.   <characteristic type="CertificateStore">
n.     <characteristic type="ROOT">
o.       <characteristic type="{hash of certificate}">
p.         <parm name="EncodedCertificate" value="{encoded hash of certificate}"/>
q.       </characteristic>
r.     </characteristic>
s.   </characteristic>
t. </wap-provisioningdoc>
```
12. Save the XML document as an ASCII file named _setup.xml.

Delivering the Certificate

You can deliver the completed provisioning file to a Windows Mobile-powered device using any one of the following means:

- Over the air using an OMA DM Server
- Over the air using an OMA Client Provisioning (formerly WAP Client Provisioning) server
- Wrapped in a .cpf file and sent by using Internet Explorer Mobile, ActiveSync, SI/SL, or Storage Card.

Note Microsoft recommends that you provision the device using over-the-air (OTA) methods when possible. If you must deliver the XML in a file, we recommend that you package and sign provisioning documents in the CAB Provisioning Format (.cpf). An XML provisioning document may not install on a Windows Mobile-powered device if the file containing the document is not signed.

Note Cabinet files and each DLL and executable within a cabinet file must be signed, including resource-only DLLs.

Installing Certificates on a Windows Mobile 6 Powered Device

In Windows Mobile 6, the Mobile Operator or Enterprise IT Professional can create a CAB file with a certificate appropriate for use within the corporation. The Manager or Authenticated user role can use this CAB file to add certificates to the user (HKCU) Root and CA stores on the device.

Note Windows Mobile 6 supports the use of wildcard certificates for device-to-server authentication.

The certificate installer on Windows Mobile 6 devices will install certificates delivered in the following file formats:

- PFX/.P12 . Public-Key Cryptography Standards #12 (PKCS #12) format files that include personal certificates with private keys as well as certificates that install into the intermediate and root certificate stores.
- CER . Base64-encoded or DER-encoded X.509 certificates that install into the intermediate and root certificate stores.
- P7B - Public-Key Cryptography Standards #7 (PKCS #7) format files that install multiple certificates to any certificate store on the device.

The files can be delivered to the device via desktop ActiveSync, removable storage card, e-mail attachment, or Mobile Internet Explorer file download. Windows Mobile 6 Professional devices also allow download from a file share. When the file is opened from the file explorer, the certificate installer processes and installs the file automatically. The following table shows what kinds of certificates and keys the different types of files support.

File Type	Private key support	Installs a certificate chain	Installs only one certificate	Installs multiple certificates (can include chains)
.PFX/.P12	Yes	Optional	Optional	Yes
.CER	No	No	Yes	No
.P7B	No	Optional	Optional	Optional
Certificate Enroll operation	Yes	Yes	No	No

Certificate Chains

A certificate chain consists of all the certificates needed to certify the subject identified by the end certificate. In practice this includes the end certificate, the certificates of intermediate CAs, and the certificate of a root CA trusted by all parties in the chain. Every intermediate CA in the chain holds a certificate issued by the CA one level above it in the trust hierarchy. The root CA issues a certificate for itself.

Algorithm for Adding Certificate Chains

When importing the certificate for a client, the certificate chain may be included in the .PFX file. This enables the device to authenticate the intermediate and root certificates associated with the end certificate. All certificates in the chain will be added to the appropriate certificate stores on the device in order to enable trust validation.

If the chain certificates are included in the .PFX file, the application processes the chain certificates as follows:

1. Store the subject certificate in the MY certificate store. The subject certificate has a public key associated with the private key that is being added to the device as a part of the PFX import.
2. Check for existence and install any certificate that meets both of the following requirements into the ROOT Certificate store,
 - a. The certificate is self-signed by its own private key.
 - b. The issuer of the certificate is the same as the subject of the certificate.
3. Check for the existence of and install any other certificates provided in the chain (intermediate certificates) to the CA certificate store.

Certificate-based Authentication

Windows Mobile 5.0 with MSFP and later devices can use SSL with Transport Layer Security (TLS) client authentication in place of SSL basic authentication. Certificate-based authentication offers a significant security advantage over the use of single-factor password-based authentication. This advantage comes from two factors. First, the strength of the key is pre-determined by the administrator and can be very strong. Windows Mobile and Windows Server together support up to 2,048-bit keys. Second, requiring certificate-based authentication greatly reduces the risk that a user's credentials will be compromised. A user can't loan out their password, and an attacker who can sniff over-the-wire traffic won't be able to recover usable credentials.

To use certificate-based authentication with Windows Mobile, the mobile device must contain the root certificate for the Exchange front-end or client access server it's communicating with; the mobile device must also have its own client user certificate with the associated private key. The user certificate enrollment process can only occur when the device is connected to a desktop in the same domain as the enrollment web site.

Certificate-based Authentication for Windows Mobile 5.0

The certificate enrollment process for Windows Mobile 5.0 MSFP devices uses ActiveSync desktop to connect to the corporate CA to receive the required user certificate and associated key. Once the user certificate and key are on the device, cradling with Desktop ActiveSync 4.1 or later is required only to renew the certificate when it expires.

Microsoft has created a tool for deploying Exchange ActiveSync certificate-based authentication; it can be downloaded from the following Microsoft Download center Web site.

<http://go.microsoft.com/fwlink/?LinkId=54738>.

For an overview of setting up certificate-based authentication for use with Windows Mobile and Exchange ActiveSync, see Appendix A, Overview of Deploying Exchange ActiveSync with

Certificate-Based Authentication, of the Step-by-Step Guide to Deploying Windows Mobile-based Devices with Microsoft Exchange Server 2003 SP2, available at <http://www.microsoft.com/technet/solutionaccelerators/mobile/deploy/msfpdepguide.mspx>.

Certificate-based Authentication for Windows Mobile 6

With Windows Mobile 6, the process for implementing Transport Layer Security (TLS) certificate-based authentication has been streamlined and made easier to maintain. The system administrator creates a certificate type and makes it available through Active Directory. The user then authenticates to the network, navigates to the designated location, and the client user certificate with the associated encrypted private key is passed to the user's device.

Note Wildcard certificates or certificates not supplied by a Microsoft Certificate Authority Server can be used with Windows Mobile 6 powered devices.

Managing Certificates with the CertificateEnroller Configuration Service Provider

Applies to Windows Mobile 6

The CertificateEnroller Configuration Service Provider in Windows Mobile 6 devices enables you to generate certificates and associate them with a key pair to produce and install trusted certificates for your mobile devices. You can define each certificate type and publish them for other client devices and servers in your corporate network. The CertificateEnroller also provides management and certificate renewal features.

Using the CertificateEnroller Configuration Service Provider with the SECROLE_USER_AUTH role on a device, you can add, delete, or query certificates in the HKCU (User) CA and ROOT certificate stores. If SECROLE_USER_AUTH is granted the SECROLE_MANAGER on the device, you can also add certificates to the HKLM (system) certificate stores. For more information about the certificate stores on mobile devices, see Certificate Management in Windows Mobile Powered Devices.

The CertificateEnroller Configuration Service Provider allows you to

- Configure a certificate type
- Configure a certificate type and trigger device enrollment
- Securely enroll for a certificate using a pre-configured certificate type
- Query for and renew existing certificate types

The CertificateEnroller will download the full chain of certificates including the root and any intermediates by requesting the .pb7 file from the certificate server. The path to the file is specified in ServerPickupPage parameter of the CertificateEnroller Configuration Service Provider.

Note This Configuration Service Provider can be managed over both the OMA Client Provisioning (formerly WAP Client Provisioning) and the OMA DM protocol.

Using the Desktop ActiveSync, your device users can install the certificate from the corporate server to their cradled Windows Mobile-powered device. The existing corporate desktop logon procedure -- password, smartcard, or other means of user identification, authenticates the enrollment, streamlining the distribution of the certificate/encrypted key pair.

Using Desktop Enrollment

To prepare for using desktop certificate enrollment, the system administrator should do the following:

- Set up or have access to a Windows 2000, 2003 or later Windows Certificate Server.

- Create the certificate type or use an existing certificate published to Active Directory.
- Inform users of the location of the certificate on the corporate network.
- Provide users with instructions for using the **Get Device Certificate** user interface.

Once you have published the certificate to Active Directory and directed the users to enroll for the certificate, the users will step through the following process:

To enroll for a certificate with a Windows Mobile 6 powered device

1. From Advanced Tools, the user chooses Get Device Certificate, navigates to Active Directory on the corporate network, selects the desired certificate, and clicks Add.
2. The desktop processes the enrollment while the user waits a short period of time. During this time, the device generates a public/private key set and proxies the enrollment to the Windows Certificate Server through the desktop.
3. The CA returns a signed certificate to the desktop which, in turn, delivers the certificate to the device.
4. The device stores the certificate and its chain of certificates to the root CA. If the root certificate is not already in the root certificate store of the device, the user is asked to accept the certificate.
5. The user sees a success dialog to denote the end of the enrollment process.

Once the certificate is in the user Root or CA store, the mobile device will be ready to authenticate with the desired protocol.

Revoking a Certificate for a Signed Application

The certificate model allows you to stop applications from executing by revoking the certificate used to sign the application. This is helpful if a malicious signed application was installed on the device, or you do not want users to use a particular application.

With revocation, you can block a specific application or group of applications from running on the device:

- A whole class of applications or all applications from a specific ISV can be blocked by revoking a code-signing certificate or a Certificate Authority (CA) certificate.
- An individual executable or DLL can be blocked by revoking the hash of the binary.

Revoking an unsigned executable/dll is only secure against off-line modification if the device doesn't allow unsigned code to run. Otherwise, someone can use a hex-editor to change a byte in the executable, which will change the hash of the module and thus the device would allow it to run.

When revoking a certificate, you should use the thumbprint of the certificate's hash. You can use the `revoke.exe` tool in the SDK to inspect or generate the appropriate hash, then use the LoaderRevocation Configuration Service Provider to create a revocation XML.

You would send the revocation XML that has the hash for the revoked certificate to the device over the air (OTA) or by pushing it in a cab provisioning file (cpf). In both instances, the message must originate from a source that has a manager role on the device.

Security Services for Windows Mobile 5.0 and Windows Mobile 6

Windows Mobile implements the following security services as part of the core operating system.

Service	Description
Cryptographic services	<p>Cryptography helps provide privacy and authentication. Windows Mobile offers the following cryptographic services:</p> <ul style="list-style-type: none"> • Encryption, to help provide privacy and authentication between two communicating parties who have exchanged a shared secret. • Hashing, to help insure data integrity of information when sent over a nonsecure channel such as the Internet and to protect user credentials on the device. For example, with Basic Authentication, the user credentials are hashed while stored on the device. • Digital Signature, to help authenticate another party, or information sent by that party, without prior exchange of a shared secret. <p>Cryptographic algorithms are used to provide these services. The algorithm implementation is certified as compliant with the US Federal Information Processing Standard (FIPS) 140-2, level 1. This certification asserts that the Windows Mobile cryptographic implementations work properly and that they are secure against a variety of potential threats. Supported algorithms include the US Government standard Advanced Encryption Standard (AES) in 128-, 192- and 256-bit key lengths, single and triple DES, the Secure Hash Algorithm (SHA-1), and RSA public-key encryption and decryption.</p> <p>For more information about FIPS, see Cryptographic Services and FIPS Compliance in Windows Mobile 5.0 and Windows Mobile 6.</p>
Authentication services	<p>Authentication services can be used by application developers to authenticate clients. Services include security services or client certificates for user authentication, credential management, and message protection. Services include:</p> <ul style="list-style-type: none"> • Security services for user authentication • Credential management. • Message protection through a programming interface called Security Support Provider Interface. • Windows Mobile provides integrated support for remote access networking and authentication, including Windows NT® LAN Manager Challenge/Response protocol version 2 (NTLMv2), SSL 3.1, Private Communications Technology (PCT), Point-to-Point Protocol (PPP), and the Wireless Transport Layer Security (WTLS) class 2 for accessing secure Wireless Access Protocol (WAP) sites.
Virtual private networking support	<p>Built-in support for virtual private networking, using Layer Two Tunneling Protocol with Internet Protocol Security (IPSec) encryption (LT2P/IPSec) or Point-to-Point Tunneling Protocol</p>

	<p>(PPTP) in combination with strong passwords using the Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAPv2). Third-Party VPNs may be installed.</p> <p>For more detailed information about VPNs, PPTP, or IPsec/L2TP, see this Microsoft Web site. http://go.microsoft.com/fwlink/?LinkID=82573&clcid=0x409</p>
<p>Wi-Fi encryption</p>	<p>Support for the Wireless Protected Access (WPA and WPA2) and (Wireless Network Encryption Types) Wired Equivalent Privacy (WEP) encryption standards for use with 802.11a/b/g wireless LANs.</p> <p>The following are some of the product compatibility standards for wireless local area networks (WLAN) based on the IEEE 802.11 specifications:</p> <ul style="list-style-type: none"> • WEP (Wired Equivalent Privacy) provides data confidentiality services by encrypting the data sent between wireless nodes. • Wi-Fi Protected Access (WPA) provides enhanced security for wireless networks and is based on a subset of the IEEE 802.11i standard. <p>Applies to Windows Mobile 6:</p> <ul style="list-style-type: none"> • WPA2 provides a stronger encryption mechanism through Advanced Encryption Standard (AES) with key sizes of 128 and 256.
<p>Storage card encryption</p>	<p>Applies to Windows Mobile 6:</p> <p>Support for encryption of data stored in removable storage cards. Storage card encryption supports Advanced Encryption Standard (AES) in 128 bit cipher strength.</p> <p>The following list shows the storage card encryption support:</p> <ul style="list-style-type: none"> • Encrypt data written from the mobile device to removable media. The data will be encrypted for use on the encrypting device only. If unencrypted data is transferred to the storage card by another device (Phone, PC), the content is not encrypted by the device. ActiveSync file explorer provides desktop access to encrypted data files. • Enable Over-the-Air (OTA) provisioning of encryption via Exchange or other OTA DM solution. <p>OEMs and Mobile Operators can provision the encryption policy during a cold boot of the device.</p> <p>Encryption is transparent to applications and user, not including performance impacts.</p> <p>Storage card encryption can be managed by Exchange 2007 policies. The user can also manage the mobile encryption configuration through the control panel.</p>
<p>Secure Sockets Layer (SSL) support</p>	<p>Internet Information Services (IIS) and Internet Explorer Mobile implement SSL to help secure data transmission when a user connects to a server to synchronize Microsoft Exchange data, configure the Windows Mobile-powered device, or download</p>

	<p>applications.</p> <p>The SSL protocol allows Web servers and Web clients to communicate more securely through the use of encryption. When SSL is not used, data sent between the client and server is open to packet sniffing by anyone with physical access to the network.</p> <p>To authenticate using SSL, Basic or Microsoft Windows NT LAN Manager (NTLM) authentication is used. If it is necessary to support Basic authentication, for instance for Web browsers that do not support NTLM, it is recommended that SSL be used as well so that the user's password is not sent in plain text.</p> <p>For information about configuring a web server to use SSL, see the Step-by-Step Guide to Deploying Windows Mobile-based Devices with Microsoft Exchange Server 2003 SP2 at http://go.microsoft.com/fwlink/?LinkId=81200</p> <p>For information about using SSL in a network configuration, see Security within the Corporate Network.</p> <p>Applies to Windows Mobile 6:</p> <p>Advanced Encryption Standard (AES) AES is now available for SSL channel encryption. AES is the encryption standard for the U.S. Federal Government and NSA, the National Security Agency.</p> <p>Note At present, AES cannot be used with Exchange ActiveSync (EAS) because EAS is built on IIS which does not currently support AES.</p> <p>AES is available for SSL channel encryption in 128 and 256 bit cipher strengths. NSA has approved 128, 192 and 256 bit AES ciphers as sufficient to protect classified information up to the SECRET level. TOP SECRET information requires use of either 192 or 256 bit AES ciphers. With AES encryption, Windows Mobile 6 offers the same level of security approved by NSA for TOP SECRET information, the highest level of security the U.S. government requires.</p>
--	--

Windows Mobile exposes these security services so that applications can make use of them; for example, the built-in Outlook Mobile client can use SSL (and, by extension, various cryptographic algorithms) for POP and IMAP accounts.

Cryptographic Services and FIPS Compliance in Windows Mobile 5.0 and Windows Mobile 6

Windows Mobile supports the technologies required for Federal Information Processing Standard (FIPS) compliance. FIPS certification is required for selling products to the federal government. Some security-sensitive industries such as finance and insurance, have also adopted FIPS certification.

FIPS 140-1 and its successor, FIPS 140-2, are U.S. Government standards that provide a benchmark for implementing cryptographic software. They specify best practices for implementing cryptographic algorithms, handling key material and data buffers, and working with the operating system.

An evaluation process administered by the National Institute of Standards and Technology (NIST) Cryptographic Module Validation (CMV) Program (<http://csrc.nist.gov/cryptval/>) allows encryption product vendors to demonstrate the extent of their compliance with the standards, and thus the trustworthiness of their implementations.

Note NIST CMV as of May 26, 2002 accepts validation test reports for cryptographic modules against only FIPS 140-2. However, it states on the CMV program Web page that "agencies may continue to purchase, retain and use FIPS 140-1 validated products after May 25, 2002."

The following Cryptographic Service Providers have completed U.S. Government FIPS 140-2, level 1:

- Windows CE Enhanced Cryptographic Service Provider 5.0
- Windows CE Enhanced Cryptographic Service Provider 5.01 (this is the default for Windows Mobile-powered devices, and the certificate is listed in the security policy of the Windows CE certificate (<http://go.microsoft.com/fwlink/?LinkId=83398>).

Applications make use of cryptographic modules to perform cryptographic operations. As long as they use of FIPS certified crypto modules and FIPS approved algorithms, and are run on FIPS capable operating systems, then they are automatically FIPS compliant. There is no need for additional certifications.

The following table shows examples of components that are Windows CE Enhanced Cryptographic Service Provider 5.01 FIPS compliant.

FIPS Compliant	Description
Active Sync	Microsoft ActiveSync provides support for synchronizing data between a Windows-based desktop computer and Windows Embedded CE-based devices.
EAP/CHAP	Extensible Authentication Protocol (EAP) provider for MD5 Challenge Handshake Authentication Protocol (CHAP). EAP allows third-party authentication applications to interact with the Point-to-Point Protocol (PPP). CHAP is an encrypted authentication mechanism that avoids transmission of the actual password on the connection.
EAPOL	Extensible Authentication Protocol Over LAN implements the state machine for 802.1x authentication. It facilitates the sending and receiving of packets on the network and receives network status and configuration information, and registers with

	the EAP framework for provider-specific operations.
IPSec	<p>IPSEC family of protocols that can be used for IETF standard end-to-end encryption with Windows 2000, Windows XP, or Windows Server 2003 systems, including:</p> <ul style="list-style-type: none"> • L2TP/IPSec VPN client and server for remote access • L2TP/IPSec tunnels for gateway-to-gateway VPN connections • IPSec tunnels for gateway-to-gateway VPN connections
MSCHAP	Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is an encrypted authentication mechanism very similar to CHAP, except it is supported by both PPP and EAP.
NTLM	NTLM is a Windows challenge response authentication system that uses Windows domain credentials for authentication.
PPP	Point-to-Point Protocol (PPP)
RSAENH module	<p>The certification applies to RSAENH cryptographic service provider module (rsaenh.dll).</p> <p>Applies to Windows Mobile 6</p> <p>The RSAENH crypto module certified is an updated version of the module that shipped with CE 5.0.</p> <p>Windows Mobile uses the same RSAENH crypto module as CE 5.01. Hence Windows Mobile is FIPS capable. Currently the certificate does not explicitly mention Windows Mobile as a tested operational environment.</p> <p>The FIPS certificate for RSAENH can be found at this We site: http://csrc.nist.gov/cryptval/140-1/1401val2005.htm#560</p>
S/MIME protocol	S/MIME e-mail encryption protocol that is used to protect the confidentiality and integrity of e-mail messages
Schannel (SSL)	Microsoft Remote Desktop Protocol (RDP) 5.2, or later, of Terminal Service Client, which is available from Windows Server 2003 and runs on a Windows XP or later machine, connecting to a Terminal Server session on a Windows 2003 Server configured for FIPS-compatible encryption
SQL TDS	SQL Tabular Data Stream (TDS) protocol that is used with the Windows TLS/SSL Security Provider between SQL clients and SQL Server 2000, or later
TLS protocol	The IETF RFC 2246 Transport Layer Security (TLS) protocol that is used between the Web browser (Internet Explorer) and Web server (Internet Information Server)
Terminal Services Client	<p>Supports the user interface for Windows Terminal Server and Remote Desktop Protocol (RDP).</p> <p>It is software that enables a device to access Windows-based applications on the Terminal Server.</p>

Third party and end-user-developed software that requires cryptographic services can call on the CryptoAPI to invoke this cryptographic service provider.