

## Capability Descriptions

Symbian has worked closely with stakeholders from across the industry to agree the criteria to provide access to capabilities. If no capabilities are required or the developer wants to get an application signed that requires capabilities associated with the 'basic' set, then they will just need to follow the standard Symbian Signed tests.

If the developer requires access to 'extended' capabilities, they will need to run through the Symbian Signed tests and provide detailed information in a series of declarative statements. For declarative statements the developer declares which APIs associated with a capability they are using and why they're using them.

There is further set of capabilities within the extended set of capabilities called 'Phone Manufacturer Approved' capabilities. Access to these can only be authorized by a phone manufacturer or channel certifier.

### Basic Capabilities

#### LocalServices

**Access to services over 'short-link' connections (such as Bluetooth or infra-red). Such services will not normally incur cost for the user**

The location of the remote service is assumed to be well known to the user. A program with this capability can normally send or receive information through a serial port, USB, IR and point-to-point Bluetooth profiles. Examples of local services are synchronization of data with the user's PC, file transfer, etc. This capability does not allow use of IP or any routable Bluetooth profiles, or spending of a user's money by dialing a telephone number.

#### Location

**Access to data giving the location of the phone**

This capability supports the management of a user's privacy regarding the mobile phone's location.

#### NetworkServices

**Access to remote services (such as over-the-air data services or Wi-Fi network access), which might incur cost for the user**

This capability allows access to a remote service without any restriction on its physical location. Typically, this location is unknown to the user.

Voice calls, SMS and Internet services are good examples of such network services. This capability controls access to services delivered via GSM, CDMA and all IP transport protocols including IP over Bluetooth ('PAN profile').

#### ReadUserData

**Read access to confidential user data**

This capability supports the management of a user's privacy.

Typically contacts, messages and appointments are always seen as the user's confidential data. For other content, such as images or sounds, there could be a choice to be made by the user.

#### UserEnvironment

##### **Access to live data about the user and their immediate environment**

This capability protects the user's privacy.

Examples of services protected using this capability are audio, picture and video recording, and biometrics (such as fingerprint) recording. Please note that the location of the device is excluded from this capability and is instead protected by using the dedicated capability `Location`.

#### WriteUserData

##### **Write access to confidential user data**

This capability supports the management of the integrity of user data.

Please note that this capability is not always symmetric with `Read-UserData`. For instance, one might wish to prevent rogue software from deleting music tracks but not wish to restrict read access to them.

Software developers creating programs (whether system servers or applications) may use this capability to control access to their data when it is stored in private directories.

It is not always obvious whether to treat data as confidential and the choice will depend on the UI implementation.

## Extended Capabilities

#### PowerMgmt

##### **The ability to kill any process, to power-off unused peripherals and to cause the mobile phone to go into stand-by, to wake up, or to power down completely**

Note that this doesn't control access to anything and everything that might drain battery power.

#### ProtServ

##### **Allows a server process to register with a protected name**

Protected names start with a '!'. The kernel will prevent servers without `ProtServ` capability from using such a name, and, therefore, will prevent protected servers from being impersonated. All servers in the TCE have this capability.

#### ReadDeviceData

##### **Read access to confidential network operator, mobile phone manufacturer and device settings**

Settings that are not confidential (such as the system clock) do not need to be protected by this capability.

Examples of confidential device data include the list of installed applications and the device lock PIN code.

#### SurroundingsDD

##### **Access to logical device drivers that provide input information about the surroundings of the mobile phone**

Good examples of drivers that require this capability would be GPS and biometrics device drivers. For complex multimedia logical device drivers that provide both input and output functions, such as a sound device driver, the `MultimediaDD` capability should be used where it is impractical to separate the input from the output calls at its API level.

#### SwEvent

##### **The ability to simulate key presses and pen input and to capture such events from any program**

Note that, when it has the user input focus, normal software does not need `SwEvent` in order to be dispatched key and pen events.

#### **TrustedUI**

**The ability to create a trusted UI session and, therefore, to display dialogs in a secure UI environment**

Trusted UI dialogs are rare. They must be used only when confidentiality and security are critical: for instance for password dialogs.

Normal access to the user interface and the screen does not require this.

Code implementing a trusted UI dialog would need `SwEvent` capability.

Note that trusted UI dialogs are not implemented in Symbian OS v9.1.

#### **WriteDeviceData**

**Write access to settings that control the behavior of the device**

This setting is not always symmetrical with `ReadDeviceData`, i.e. just because data important to maintaining the integrity of the system is protected from being written does not mean that it needs to be protected against being read.

Examples of this type of setting are device lock settings, system time, time zone, alarms, etc.

## **Phone Manufacturer Approved Capabilities**

#### **AllFiles**

**Read access to the entire file system and write access to other processes' private directories**

Similarly to `Tcb`, this capability is very strictly controlled and it is not granted lightly. Nevertheless, phone manufacturers' test software might reasonably have it.

For instance, the backup and restore server might need it to backup data on behalf of programs.

Unlike `Tcb`, `AllFiles` permits read and write in `\private`.

The system capability `AllFiles` can be used in the following circumstances:

- By mobile phone manufacturers wishing to have a powerful shell or file manager. In this case, the user would be allowed to destroy or modify some servers' private files. Symbian therefore highly discourages such a facility being made publicly available
- For test utilities to retrieve files in order to audit them to validate the behavior of a subsystem.

#### **CommDD**

**Direct access to all communications equipment device drivers**

This includes for example, WiFi, USB and serial device drivers.

#### **DiskAdmin**

**Access to file system administration operations that affect more than one file or directory (or overall file-system integrity and behavior, etc.)**

This includes, for example, mounting and unmounting a drive partition.

#### **Drm**

**Access to DRM-protected content**

DRM agents use this capability to decide whether or not a program should have access to protected content. Symbian OS trusts that software granted `Drm` capability will respect the rights associated with this content.

#### **MultimediaDD**

**Access to critical multimedia functions such as direct access to associated device drivers and priority access to multimedia APIs**

This includes sound, camera, video, etc.

**NetworkControl**

**The ability to modify or access network protocol controls**

Typically when an action can change the behavior of several existing and future connections, it should be protected by `NetworkControl`.

For example, forcing all existing connections on a specific protocol to be dropped or changing the priority of a call.

**Tcb**

**Write access to executables and shared read-only resources**

`Tcb` allows write access to `\sys` and `\resource` directories. This is the most critical capability as it allows write access to executables, which contain the capabilities that define the security attributes of a process.

No third-party code should be allowed to do this. The TCB processes run with at least this capability.

**This material is taken from the book *Symbian OS Platform Security* from Symbian Press. More information on this book and the other books in the Symbian Press series is available from <http://developer.symbian.com/main/academy/press/index.jsp>**