Automatic Setup of Intermec CN3

Professional Services
Intermec UK
INTRODUCTION ..................................................................................................................... 3

SCRIPTING ON INTERMEC DEVICES .............................................................................. 4
  SMARTSYSTEMS TRANSFER AGENT .................................................................................. 4
  _sstransferagent.xml Structure ......................................................................................... 4
  _sstransferagent.xml Elements ......................................................................................... 4
  _sstransferagent.xml Locations ......................................................................................... 6
  Transfer Agent Logging ........................................................................................................ 7
  AUTO TOOLS .................................................................................................................... 8
  AutoExec .......................................................................................................................... 8
  AutoRun ............................................................................................................................ 8

INSTALLING DEVICE SETTINGS ...................................................................................... 9
  INTERMEC SETTINGS ....................................................................................................... 9
    Determining Intermec Settings ......................................................................................... 9
    Applying Intermec Settings ............................................................................................. 10
  MICROSOFT WINDOWS MOBILE CONFIGURATION MANAGER SETTINGS .................... 12
    Determining Microsoft Windows Mobile Configuration Manager Settings .................. 12
    Applying Microsoft Settings .......................................................................................... 12
  REGISTRY SETTINGS ....................................................................................................... 14
    Applying Registry Settings ............................................................................................. 14

OTHER FEATURES .......................................................................................................... 15
  INSTALLING CAB FILES IN SPECIFIC ORDER .................................................................. 15
  REBOOTING AUTOMATICALLY AFTER CAB FILE INSTALLATION .................................... 15
  SERVICE RELEASE INSTALLATION .................................................................................. 15

EXAMPLES ...................................................................................................................... 16
  EXAMPLE 1 ....................................................................................................................... 16
  EXAMPLE 2 ....................................................................................................................... 16
  EXAMPLE 3 ....................................................................................................................... 17
  EXAMPLE 4 ....................................................................................................................... 18
  EXAMPLE 5 ....................................................................................................................... 18

APPENDIX A – CONFIGURING GPRS CONNECTIONS ....................................................... 19
  PROVISIONING XML FOR GPRS CONNECTION .......................................................... 19
  “Always On” Connections ............................................................................................... 20

APPENDIX B – TYPES OF REBOOT .................................................................................. 21
  WARM BOOT .................................................................................................................. 21
  COLD BOOT .................................................................................................................. 21
  CLEAN BOOT ................................................................................................................ 21

REVISION HISTORY ........................................................................................................ 22
Introduction

It is a common requirement to set up a rugged mobile device so that applications and settings are automatically installed without user intervention. Setting up a device in this way can reduce the time taken to stage and deploy new devices. It also avoids errors that can occur with manual configuration. In addition it allows a device to be reset back to a known state if a problem should arise in the field.

This document explains various techniques that can be used to accomplish this on the Intermec CN3. It explains how to create a set of files that can automatically install software and apply settings. These files are placed into one (or both) of the storage areas on the CN3 that are not deleted by a clean boot. These are the internal "Flash File Store" and the miniSD card. Once these files are in place, a clean boot is all that is required to prepare the device for use.

The examples accompanying this document demonstrate these techniques, taking a device from the factory default state to running an application without any user intervention.

This document does not assume or require that device management software is being used. However, it is possible to deploy the files described in this document via device management software to further automate the device setup process.
Scripting on Intermec Devices

SmartSystems Transfer Agent

When a clean boot is performed on the CN3, the Intermec SmartSystems Transfer Agent on the device will process one or more XML script files to perform actions such as copying files & directories and running executables. These script files are always named _sstransferagent.xml.

Following a clean boot, Transfer Agent will execute each _sstransferagent.xml file once. It will not be executed on subsequent reboots until the device is clean booted again.

_sstransferagent.xml Structure

The structure of a typical _sstransferagent.xml file is shown below:

```xml
<?xml version="1.0"?>
<Devices>
  <Device Type="" Family="" Model="" Boot="">
    <Files SrcDir="">
      <File SrcName="" DestDir="" DestName="" OS="" ConfigString="" Run="" CmdLine="" StartIn="/"/>
    </Files>
    <Commands>
      <DeleteFile File="/"/>
      <DeleteDir Dir="/"/>
      <CopyDir SrcDir="/" DestDir="/"/>
      <SetRegKey KeyName="/" ValueName="/" Value="/" Type="/"/>
    </Commands>
  </Device>
</Devices>
```

_sstransferagent.xml Elements

<Devices> Element

The Transfer Agent XML file consists of a <Devices> element containing one or more <Device> elements.

<Device> Element

The <Device> element can contain <Files> and/or <Commands> elements.

The <Device> element has four attributes: Type, Family, Model and Boot.

Type, Family, and Model are derived from the first three or four characters of the hardware configuration string (also known as the Configuration Number or CN) of the device as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>HardwareConfig[0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Family</td>
<td>HardwareConfig[1]</td>
</tr>
<tr>
<td>Model</td>
<td>HardwareConfig[2-3]</td>
</tr>
</tbody>
</table>

Transfer Agent will only process <Device> elements that match the current device. * can be used as a wildcard if desired.
Boot is optional, and can be set to "Warm" or "Cold". This can be used to reboot the device after the script has been processed.

<Files> Element
The <Files> element contains a list of <File> elements that define files to copy or execute.

<File> Element
The <File> element can be used to copy files or run executables.

Several attributes are used to copy a file. SrcName is the name of the file to copy from the directory specified in the <Files> SrcDir parameter. DestName specifies the new name of the file when copied into destination directory specified by DestDir.

The optional ConfigString attribute allows you to place conditions on whether a file is copied or not, based on the hardware configuration of the device. It's a mask that allows the use of "?" and "*" wildcards. The '?' wildcard matches any character in a single position. The '*' wildcard matches every character to the end of the configuration string, but it cannot be used in the middle of the string, e.g.

- CN3*     copy file on any CN3 device
- CN3?????1* copy file on CN3 with internal GPS option

The OS attribute also allows conditional copying. On the CN3 this should always be set to "PocketPC".

Another form of the <File> element can be used to run an executable. In this case, SrcName is not used and should be left blank. DestDir and DestName identify the location and name of the executable to run. The Run attribute specifies whether script processing waits until the executable has finished:

- True          run executable and wait for it to exit before performing next command
- NoWait        run executable and continue to next command without waiting

CommandLine specifies the command-line parameters, if any, to the executable named in DestName. If you need to use quotes in the command-line itself, enclose the value for the CommandLine parameter in single quotes. For example, to run this command:

```
\Flash File Store\MySetupApp.exe /PATH="\Flash File Store\My Setup Files"
```

you would use this line:

```
<File SrcName="" DestName="MySetupApp.exe" DestDir="\Flash File Store" ConfigString="CN3*" Run="true" CommandLine="/PATH="\Flash File Store\My Setup Files"" StartIn="\Flash File Store" />
```

StartIn specifies the current directory though this is ignored on Windows Mobile devices.

<Commands> Element
The <Commands> element contains one or more <DeleteFile>, <DeleteDir>, <CopyDir>, <SetRegKey> and/or <Chain> elements.
<DeleteFile> Element
The <DeleteFile> element will delete a file. The file is specified with the File attribute.

<DeleteDir> Element
The <DeleteDir> element will delete a directory and all its contents. The directory is specified with the Dir attribute.

<CopyDir> Element
The <CopyDir> element will copy a directory and all its contents. The attributes SrcDir and DestDir specify the source and destination directories.

<SetRegKey> Element
The <SetRegKey> element can be used to set a registry value under HKEY_LOCAL_MACHINE. It has four attributes:

- KeyName: subkey name under HKEY_LOCAL_MACHINE
- ValueName: name of the value under KeyName
- Value: value to be set
- Type: value type which defaults to string but can optionally be DWORD

<Chain> Element
The <Chain> element has a single attribute name Dir. This specifies the directory name to prepend to the _sstransferagent.xml file name before trying to parse again. This is used to pass control to another _sstransferagent.xml script.

_sstransferagent.xml Locations

Flash File Store SSPB Directory
By default, the only _sstransferagent.xml script present on the CN3 is:

\Flash File Store\SSPB\_sstransferagent.xml

This is used to install Intermec value-added components known as the SmartSystems Platform Bundle (SSPB). These are the CAB files you see installing when you clean boot a CN3. If you examine this script you will see it does the following:

- Copies SSPB CAB files from \Flash File Store\SSPB into \CabFiles
- Copies various other files from SSPB folder into correct locations
- Installs itcauto.cab which creates the \2577 folder containing the "Auto Tools"
- Executes \Windows\RunAutoRun.exe to run AutoExec and AutoRun

The CAB files copied into \CabFiles directory are then installed by AutoCab which is called from AutoExec.dat. The CAB files are deleted from \CabFiles after installation.

This file can be edited to disable components of the SSPB. This is what the InstallSelect tool does. It modifies the ConfigString attribute of the <File> element by placing an "X" in front of the configuration string. This means it no longer matches the configuration string of the device and so the component is not loaded.
Root of Flash File Store or SD Card

An \_sstransferagent.xml script placed in one of these locations will take precedence over the default script in the SSPB folder:

\Flash File Store\_sstransferagent.xml
\SD Card\_sstransferagent.xml

For example, the SD card used to install an updated SSPB has an _sstransferagent.xml file in the root directory of the card. Here is the version used on the SSPB 5.50.23.0362 upgrade SD card:

```xml
<?xml version="1.0"?>
<Devices>
    <Device Type="C" Family="N" Model="3">
        <Commands>
            <Chain Dir="\Flash File Store\SSPB" />
            <DeleteDir Dir="\Flash File Store\SSPB" />
            <CopyDir SrcDir="\SD CARD\SSPB" DestDir="\Flash File Store\SSPB"/>
            <SetRegKey
                KeyName="Software\Intermec\Devmgmt\SSTransferAgent"
                ValueName="PostBootFileLocation"
                Value="\Flash File Store\SSPB"
            />
        </Commands>
    </Device>
</Devices>
```

It does the following:

- Tells Transfer Agent to run \_sstransferagent.xml in \Flash File Store\SSPB when current script is completed
- Deletes existing \Flash File Store\SSPB folder
- Copies new SSPB folder from SD card to Flash File Store
- Sets PostBootFile registry value used by Transfer Agent

A similar script can be used to perform actions before the SSPB is installed. If you create your own script then it must also use the \Chain command to pass control over to the default SSPB \_sstransferagent.xml to install SSPB afterwards.

Flash File Store UserAutoInstall directory

Starting with OS release 3.25.15.0141, the Transfer Agent will also look for a script in this location:

\Flash File Store\UserAutoInstall\_sstransferagent.xml

This gives the user a way of using the Transfer Agent scripting facilities without having to alter the existing \_sstransferagent.xml file in the SSPB folder.

Note that this file is processed after the SSPB has been installed and initialized for the first time i.e. after the first automatic warm boot following a clean boot.

Transfer Agent Logging

The Transfer Agent creates a log file of its actions in a text file called SSTransferAgent.txt in the root directory of the device. This is a useful source of troubleshooting information if your scripts are not working as you expected.
Auto Tools

One of the CAB files in the SSPB is itcauto.cab. This installs several executables and scripts used to initialize the device. These are installed into the \2577 directory.

The EXE files are used to perform various tasks when the device starts up. For more information on these tools and their command-line parameters, please see the section titled “Launching Your Application Automatically” in the "Intermec Developers Library Resource Kit Developer’s Guide" which can be downloaded using the link below:


The DAT files are text files containing scripts executed by the corresponding EXE file. The commands used these scripts files are also documented in the IDL Developers Guide.

AutoExec

AutoExec.exe is started by \Windows\RunAutoRun.exe. RunAutoRun is executed every time the device reboots, as well as during SSPB installation.

AutoExec.dat performs various setup functions for Intermec components and should not be modified. In particular, AutoExec.dat will execute AutoCab.exe to install CAB files located in the \CabFiles directory. Remember that the SSPB installation copies CAB files from the SSPB folder in Flash File Store into \CabFiles. The CAB files are deleted from \CabFiles after installation.

AutoRun

AutoRun.exe is also started by RunAutoRun.exe and runs after AutoExec.exe. Like AutoExec.dat, AutoRun.dat should not be modified by users. However, the last command in AutoRun.dat is:

CALL "\2577\AutoUser.dat"

This calls an optional script called AutoUser.dat which, if required, can be created by the user and placed into \2577 directory to execute commands after every warm or cold boot.

A summary of the commands available in AutoUser.dat scripts:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN</td>
<td>Executes program and continues without waiting</td>
</tr>
<tr>
<td>EXEC</td>
<td>Executes program and waits for it to complete (up to 10 minutes)</td>
</tr>
<tr>
<td>CALL</td>
<td>Processes specified file of commands and returns</td>
</tr>
<tr>
<td>CHAIN</td>
<td>Processes specified file of commands and does not return</td>
</tr>
<tr>
<td>;</td>
<td>Comment line</td>
</tr>
</tbody>
</table>

It is worth noting that AutoUser.dat is executed after Intermec components have been installed and started. So, if you have an application that relies on Intermec services such as the Data Collection Engine for barcode scanning, you might consider starting your application via a RUN command in AutoUser.dat rather than the typical shortcut in \Windows\Startup, e.g.

;This is the \2577\AutoUser.dat script to start my application
RUN "\Program Files\My Company\My Application\MyApp.exe"

Also note that when running on Windows Mobile 6.1 the SmartSystems splash screen is not shown while components are being initialized for the first time. It may appear that nothing is happening after the first automatic warm boot following a clean boot, but the user _sstransferagent.xml and/or AutoUser.dat scripts will eventually be executed.
Installing Device Settings

Device settings often need to be applied automatically as well. This section explains various methods for determining and applying Intermec Settings, Microsoft Configuration Manager settings and general registry settings.

Intermec Settings

All settings found in the Intermec Settings applet on the device can also be configured using Smart Systems XML files. For more information on these files, please see the help included with the "Device Management Resource Kit" available from the "Intermec Developers Library" at:

http://www.intermec.com/idl

For example, here is how you would enable the Interleaved 2 of 5 barcode symbology with SmartSystems XML:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<DevInfo Action="Set" Persist="true">
  <Subsystem Name="Data Collection">
    <Group Name="Scanners" Instance="0">
      <Group Name="Symbologies">
        <Group Name="Interleaved 2 of 5">
          <Field Name="Enable I 2 of 5">1</Field>
        </Group>
      </Group>
    </Group>
  </Subsystem>
</DevInfo>
```

Determining Intermec Settings

Once you have manually configured the Intermec Settings on a device using the Intermec Settings applet on the device, or remotely via the Intermec Settings utility in SmartSystems Foundation console on a PC, you can retrieve the XML used to automatically apply those settings in a number of ways.

SmartSystems Foundation is a software platform that provides a single, integrated environment for hands-free deployment and management of devices located anywhere in the enterprise. It can be downloaded for free from the Intermec website.

http://www.intermec.com/smartsystems

SmartSystems Backup File

If you have SmartSystems Foundation installed, you can make a backup of the device settings. This will create a file called Settings.xml in a folder in the Software Library. The path will vary depending on device name, which by default is model name + serial number, e.g.

C:\Program Files\Intermec\SmartSystem\SS_Lib\Software\SSBack_CN3A24700700996

You can either use this file "as is" or copy the fields you want to set into another XML document. The settings are arranged in a DevInfo/Subsystem/Group/Field hierarchy which must be retained in the new file.
Intermec Settings "View Field XML"

If you have SmartSystems Foundation installed, you can view the XML for a particular setting using the Intermec Settings utility. Right-click on the icon representing your device in the console then select "Intermec Settings" from the context menu. Navigate to the setting you are interested in, then right-click and select "View Field XML" to see a complete XML file that can be used to apply that setting.

Backup File on Device

The following command-line can be used to generate an XML settings file on the device itself:

```bash
\windows\iccu.exe /P\Windows\itcReaderDataModel.xml /ip=127.0.0.1 /B=\IntermecSettings.xml
```

The Windows Mobile shortcut file "Backup Settings.lnk" that accompanies this document will execute this command for you. Note that it takes around 40 seconds to complete, at which point you will find all Intermec Settings stored in \IntermecSettings.xml on your device.

Applying Intermec Settings

Once you have the required XML to apply your settings, you can deploy it in one of several ways

SSConfigDir Directory

The SmartSystems client on the device periodically checks the following directory for SmartSystems XML files:

```
\SmartSystems\SSConfigDir
```

XML files found in this directory are processed and deleted afterwards. You can copy XML files into this folder during setup to configure the device. Please note that files placed in this folder before the SmartSystems client has installed and initialized are ignored.

BootSettings Registry Setting

You can also apply Intermec Settings every time the device boots. To do this, rename your Intermec Settings XML file to BootSettings.xml and place it in the SmartSystems folder e.g.

```
\SmartSystems\BootSettings.xml
```

You also need to set the following registry value to enable boot settings:

```
[HKEY_LOCAL_MACHINE\Software\Intermec\SSClient]
BootSettings=dword:1
```
The following _sstranferagent.xml script shows one way to set this up:

```xml
<?xml version="1.0"?>
<Devices>
  <Device Type="C" Family="N" Model="3" Boot="Warm">
    <Files SrcDir="\Flash File Store\UserAutoInstall">
      <File SrcName="BootSettings.xml" DestName="BootSettings.xml" DestDir="\SmartSystems" />
    </Files>

    <Commands>
      <SetRegKey KeyName="Software\Intermec\SSClient" ValueName="BootSettings" Value="1" Type="DWORD"/>
    </Commands>
  </Device>
</Devices>
```
Microsoft Windows Mobile Configuration Manager Settings

The Microsoft Windows Mobile Configuration Manager takes provisioning XML and passes it to various Configuration Service Providers (CSPs). Windows Mobile has a variety of CSPs covering a range of settings, including Connection Manager CSPs such as CM_GPRSEntries. The provisioning XML can be delivered to the Configuration Manager in several different ways.

For more information, please see MSDN:


Determining Microsoft Windows Mobile Configuration Manager Settings

Documentation on the available CSPs along with the provisioning XML they accept can be found in MSDN:


You can also use the RapiConfig.exe tool from Microsoft to query and test provisioning XML settings.

An example of the Microsoft provisioning XML used to set up a GPRS connection entry can be found in Appendix A.

Applying Microsoft Settings

CAB File

Microsoft XML settings can be added to a CAB file using the /prexml or /postxml switches to Microsoft's CabWiz.exe tool e.g.

CabWiz.exe "MyApp.inf" /postxml "MicrosoftSettings.xml"

Note that the <wap-provisioningdoc> element should be omitted from the XML file when added to a CAB file in this way. The resulting CAB file can then be installed automatically via AutoCab.

CPF File

To create a CPF (CAB Provisioning Format) file, you must first create an XML file called _setup.xml containing your provisioning XML. You then use the MakeCab utility from the Smart Device SDK tools folder in Visual Studio 2005 or 2008 to create the CPF file. The command used to create a .cpf file is:

makecab.exe _setup.xml myFile.cpf

More information on CPF files can be found here:


You can test a CPF file on the device by tapping on it in File Manager, just like a CAB file. However, AutoCab does not install CPF files. You must instead use wceload.exe to install:

wceload.exe "\Flash File Store\MySettings.cpf" /silent /verifyconfig /nodelete
For more information on wceload.exe see:


This can be scripted from the user _sstransferagent.xml or AutoUser.dat script file.
Registry Settings

Many other settings on the device can be configured via the Registry. A warm boot is typically required to apply changes made to Registry settings.

Applying Registry Settings

CAB File

Registry settings can be placed into CAB files using "AddReg" section(s) in the CabWiz INF file used to build the CAB file. For more information see MSDN:


AutoReg

The Intermec utility AutoReg.exe, one of the executables installed into \2577 by itcauto.cab, can be used to import and export REG files. AutoReg is documented in the "Intermec Developers Library Resource Kit Developer's Guide".

Here is an example of how you would run this using a File command in _sstransferagent.xml:

<File SrcName="" DestName="AutoReg.exe" DestDir="\2577" Run="true" CmdLine='"\Flash File Store\UserAutoInstall\RegistrySettings.reg"' />

_sstransferagent.xml

The Transfer Agent has limited registry editing capability. You can create a string or DWORD value under HKEY_LOCAL_MACHINE via a SetRegKey element in an _sstransferagent.xml file. Here is an example:

<SetRegKey KeyName="Software\My Company\My Application" ValueName="My Setting" Value="42" Type="DWORD"/>
Other Features

Installing CAB files in specific order
By default AutoCab will install CAB files in an arbitrary order. If you need to set the order of installation, create a text file called AutoCab.dat with the names of the CAB files in the order you wish to install them and place AutoCab.dat into \2577 directory. Note that CAB files not listed in AutoCab.dat will still be installed, in an arbitrary order after those listed have been installed.

Rebooting automatically after CAB file installation
AutoCab can optionally perform a warm boot after CAB file installation. To enable this, at least one the CAB files should install a text file in the following location:

\Windows\__resetmeplease__.txt

Note there are two underscores on both side of resetmeplease i.e. four underscores in total.

When AutoCab has completed installation of CAB files, it checks for this file and warm boots device if found. Note that AutoCab must be run with –CHKRST=1 option for this to work.

Service Release Installation
A new feature added to the Transfer Agent in OS 3.25.15.0141 is the ability to automatically install Service Release CAB files. This does not require additional scripting. You simply place the SR CAB files into this folder:

\Flash File Store\SSPB\SRs

Again, these are copied and installed after the SSPB installation and initialization has completed after the first automatic warm boot following a clean boot.
Examples

The first four examples all install an application called TestApp.exe which has been packaged into TestApp.cab. It will be started automatically when the device boots using a RUN command in an AutoUser.dat script.

In addition, examples 3 and 4 install a variety of device settings using the various techniques described in this document.

Example 5 demonstrates how to automatically install Service Release (SR) CAB files.

Each example comes in two versions, one for use with internal Flash File Store and one for use on a miniSD card. To try out an example, simply copy the files and directories inside the “Flash File Store” or “SD Card” example directory into the corresponding location on the device, retaining the sub-directory structure. If using a card reader on a PC to create the miniSD card, copy the files from the “SD Card” example directory to the root of the miniSD card.

Example 1

This example applies to all versions of the CN3 OS.

Here are the files for the Flash File Store version:

```plaintext
\Flash File Store\_sstransferagent.xml
\Flash File Store\AutoUser.dat
\Flash File Store\UserCabFiles\TestApp.CAB
```

When the device is clean booted, the following happens:

- `\Flash File Store\_sstransferagent.xml` is executed
  - Copies `AutoUser.dat` into `\2577`
  - Copies all CABs from `\Flash File Store\UserCabFiles` into `\CabFiles`
  - Chains `\Flash File Store\SSPB\_sstransferagent.xml` to install SSPB
- `\Flash File Store\SSPB\_sstransferagent.xml` is executed to install SSPB
  - User CAB files also installed as they have been placed in `\CabFiles`
- Device warm boots at end of SSPB installation
- SmartSystems initialization occurs
- TestApp is started by RUN command in AutoUser.dat

Example 2

This example applies to version 3.25.15.0141 or later of the CN3 WM5.0 OS and all versions of the WM6.1 OS.

Here are the files for the Flash File Store version:

```plaintext
\Flash File Store\UserAutoInstall\_sstransferagent.xml
\Flash File Store\UserAutoInstall\AutoUser.dat
\Flash File Store\UserCabFiles\TestApp.CAB
```
When the device is clean booted, the following happens:

- \Flash File Store\SSPB\_sstransferagent.xml is executed to install SSPB
- Device warm boots at end of SSPB installation
- SmartSystems initialization occurs
- \Flash File Store\UserAutoInstall\_sstransferagent.xml is executed
  - Copies AutoUser.dat into \2577
  - Copies all CABs from \Flash File Store\UserCabFiles into \CabFiles
- Device warm boots at end of user _sstransferagent.xml due to Boot="Warm" option
- AutoCab installs user CAB files previously copied to \CabFiles
- SmartSystems initialization occurs
- TestApp is started by RUN command in AutoUser.dat

The SD card version has an additional _sstransferagent.xml located in the root of the SD card. This does the following:

- Copies \SD Card\UserAutoInstall\_sstransferagent.xml to \Flash File Store\UserAutoInstall
- Chains \Flash File Store\SSPB\_sstransferagent.xml to start SSPB installation
- Continues as above for Flash File Store version

**Example 3**

This example applies to version 3.25.15.0141 or later of the CN3 WM5.0 OS and all versions of the WM6.1 OS.

Here are the files for the Flash File Store version:

- \Flash File Store\UserAutoInstall\_sstransferagent.xml
- \Flash File Store\UserAutoInstall\AutoUser.dat
- \Flash File Store\UserCabFiles\Settings.CAB
- \Flash File Store\UserCabFiles\TestApp.CAB

This example is similar to Example 2 with the addition of a CAB file called Settings.cab in UserCabFiles directory. This CAB file configures Intermec Settings, Microsoft Settings and general registry settings. The files used to build Settings.cab can be found in the SettingsCabFile directory. They are:

- __resetmeplease__.txt
- BuildCab.bat
- IntermecSettings.xml
- MicrosoftSettings.xml
- Settings.CAB
- Settings.inf

BuildCab.bat uses Microsoft's CabWiz.exe, available in Visual Studio 2005 and 2008, to build the CAB file based on Settings.inf. The CAB file does the following:

- Creates registry settings defined in [AddRegistry] section of INF file
- Copies __resetmeplease__.txt to \Windows to request AutoCab warm boot after installation. This is required to apply registry settings.
- Copies IntermecSettings.xml to \SmartSystems\SSConfigDir
- Installs MicrosoftSettings.xml that are merged into CAB file using /postxml option of CabWiz

When the device is clean booted, the process is the same as for Example 2 with the addition of a further warm boot after Settings.cab is installed, due to the inclusion of __resetmeplease__.txt.

An alternative version of Settings.cab can be found in "BootSettings Method" subdirectory. As the name suggests, this applies Intermec Settings every time the device boots using the BootSettings XML file and registry value described elsewhere in this document.

**Example 4**

This example applies to version 3.25.15.0141 or later of the CN3 WM5.0 OS and all versions of the WM6.1 OS.

Here are the files for the Flash File Store version:

```
\Flash File Store\UserAutoInstall\_sstransferagent.xml
\Flash File Store\UserAutoInstall\AutoUser.dat
\Flash File Store\UserAutoInstall\IntermecSettings.xml
\Flash File Store\UserAutoInstall\MicrosoftSettings.cpf
\Flash File Store\UserAutoInstall\RegistrySettings.reg
\Flash File Store\UserCabFiles\TestApp.CAB
```

This example does the same as Example 3, but uses alternative methods to apply the various settings. These are achieved with additional commands in _sstransferagent.xml and settings files placed in UserAutoInstall.

- MicrosoftSettings.cpf is installed using wceload.exe
- RegistrySettings.reg is installed using Intermec AutoReg.exe
- IntermecSettings.xml is renamed to BootSettings.xml
- BootSettings registry value set to import Intermec Settings on every reboot

The files used to build MicrosoftSettings.cpf can be found in the SettingsCpfFile subfolder.

**Example 5**

This example applies to version 3.25.15.0141 or later of the CN3 WM5.0 OS and all versions of the WM6.1 OS.

Here are the files for the Flash File Store version:

```
\Flash File Store\SSPB\SRs\SR00000000_TEST_CN3_ALL.CAB
```

This example shows where SR CAB files should be placed to enable automatic installation after a clean boot.

The SD card version has an additional _sstransferagent.xml file to copy SR CAB files from \SD Card\SSPB\SRs to \Flash File Store\SSPB\SRs before chaining the default SSPB installation.

The SRTestCabFile folder contains the files required to build the test SR CAB file. All it does is install a text file into \Temp.
Appendix A – Configuring GPRS Connections

This section explains how to create the required provisioning XML to configure GPRS connections on Windows Mobile devices. See "Microsoft Windows Mobile Configuration Manager Settings" for information on how to automatically apply these settings to your device.

Provisioning XML for GPRS Connection

The CSP used to configure GPRS connections is CM_GPRSEntries. A description of the settings available for this CSP can also be found in MSDN:


There are a large number of parameters that can be set, but a simple provisioning XML file to set APN, username and password looks like this:

```xml
<wap-provisioningdoc>
  <characteristic type="CM_GPRSEntries">
    <characteristic type="O2 UK Mobile Web">
      <parm name="DestId" value="{ADB0B001-10B5-3F39-27C6-9742E785FCD4}"/>
      <parm name="Phone" value="~GPRS!mobile.o2.co.uk"/>
      <parm name="UserName" value="faster"/>
      <parm name="Password" value="password"/>
      <parm name="Enabled" value="1"/>
      <characteristic type="DevSpecificCellular">
        <parm name="GPRSInfoAccessPointName" value="mobile.o2.co.uk"/>
      </characteristic>
    </characteristic>
  </characteristic>
</wap-provisioningdoc>
```

- **DestId** is set to the GUID of "My ISP”. GUIDs commonly used when configuring Connection Manager settings are:
  - My Work Network: {18AD9FBD-F716-ACB6-FD8A-1965DB95B814}
  - My ISP: {ADB0B001-10B5-3F39-27C6-9742E785FCD4}
  - Work: {A1182988-0D73-439E-87AD-2A5B369F808B}
  - The Internet: {436EF144-B4FB-4863-A041-8F905A62C572}

- **Phone** has a special format for GPRS. Instead of a phone number, the prefix ~GPRS! is followed by the GPRS Access Point Name (APN). This is for display purposes only – the APN is actually configured in the DevSpecificCellular subsection.

- **Username** and **Password** are the credentials required to connect to the APN

- Setting **Enabled** to 1 sets this entry as the currently selected connection

- **GPRSInfoAccessPointName** sets the GPRS APN
"Always On" Connections

If you add the following line to your GPRS entry, Connection Manager will treat it as an "always on" connection. It will attempt to establish this connection as soon as the connection entry is configured, and also when the device reboots.

```xml
<parm name="AlwaysOn" value="1"/>
```

For more information see http://msdn2.microsoft.com/en-us/library/aa455988.aspx
Appendix B – Types of Reboot

Warm Boot
This is the preferred method of rebooting the CN3. Recent changes to files and the Registry will be flushed to storage memory and hardware such as radios will be gracefully powered down before the reboot occurs.

To warm boot, press and hold the power button for approximately 10 seconds.

Cold Boot
A cold boot should only be performed if the CN3 does not respond to the warm boot method. This is because cold booting the CN3 does not guarantee that cached disk data will be saved, so data in that cache may be lost during the reset. All other data is preserved.

To cold boot, first suspend the CN3 (if possible) by pressing the power button, then remove the battery, use the stylus to press the reset button in the battery compartment and replace the battery.

Clean Boot
A clean boot will reset the file system and registry to factory defaults. Only files in the "Flash File Store" and "SD Card" folders will survive a clean boot. All other files not included in the OS image will be lost.

The CN3 can be clean booted by performing a cold boot with an additional step of holding down the power button as soon as the battery is replaced and the device starts up. Keep the power button held down until the clean boot warning message appears then press one of the right side buttons to confirm the clean boot.
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-01-12</td>
<td>Initial draft release for comments</td>
</tr>
<tr>
<td>2009-01-26</td>
<td>Second draft release for comments</td>
</tr>
<tr>
<td>2009-02-06</td>
<td>First public release</td>
</tr>
<tr>
<td></td>
<td>Corrected paths in section &quot;Root of Flash File Store or SD Card&quot;</td>
</tr>
</tbody>
</table>