Windows Embedded 8

WINDOWS 8 EMBEDDED LOCKDOWN — ВОЗМОЖНОСТИ ДЛЯ ВСТРАИВАНИЯ

СЕРГЕЙ АНТОНОВИЧ
info@quarta.ru

В предыдущей статье цикла публикаций, посвященных новейшей встраиваемой ОС Windows Embedded 8, были перечислены особенности платформы Windows Embedded в общем и представлены средства разработки ОС Windows Embedded 8 Standard. Данный материал посвящен расширенным возможностям Windows 8 Embedded Lockdown для встраивания в устройства.

Одной из особенностей, отличающих Windows Embedded от классических систем Windows, являются расширенные возможности блокировки (lockdown) устройства.

Блокировка устройства подразумевает реализацию его контролируемого поведения для конечного пользователя за счет ограничения путей взаимодействия с устройством. Существуют различные причины для блокировки, например защита от проникновения в систему пользователем с терминала, определенное поведение системы для пользователя, увеличение надежности работы системы.

Windows Embedded 8 Standard основана на Windows 8, поэтому включает базовые возможности блокировки классической Windows 8: AppLocker, брандмауэр, групповые политики. Но для встраивания системы зачастую бывают необходимы дополнительные возможности. Например, если устройство представляет собой электронную кассу, нажатия комбинаций клавишей <Alt+F4>, <Alt+Tab> крайне нежелательны, так как позволят выйти за пределы специализированного приложения и получить доступ к системе.

В основном, именно эти, специфические для Embedded возможности отличают Windows семейства Embedded от классических систем Windows общего назначения:

- фильтры записи (Write Filters) и связанная с ними технология многократного восстановления системы из единожды инициированного спящего режима (Hibernate-Once-Resume-Many, HORM);
- фильтр реестра (Registry Filter);
- фильтр диалоговых окон (Dialog Filter);
- фильтр клавиатуры (Keyboard Filter);
- фильтр жестов (Gesture Filter);
- брендирование (Branding).

На рис. 1 показаны компоненты в каталоге Windows Embedded, предоставляющие возможности блокировки.

ФИЛЬТРЫ ЗАПИСИ

Фильтры записи используются во встраиваемой системе, чтобы защитить носитель данных от записи на него. Под носителем данных подразумевается любое устройство для хранения данных, поддерживаемое Windows 8. Такая возможность может быть полезна, например, для того, чтобы предотвратить многократную перезапись данных на твердотельный жесткий диск, тем самым увеличив срок его службы. Фильтры записи позволяют также представить для операционной системы носитель только для чтения как записываемый.

Включенный фильтр записи перехватывает попытки записи на защищенные носители и перенаправляет их в специальную область — оверлей, в которой сохраняются только изменения, внесенные при попытках записи на защищенный носитель. Обычно оверлей размещается в системной памяти, хотя возможно его размещение и на диске.

Оверлей в памяти полезен, когда необходимо уменьшить количество

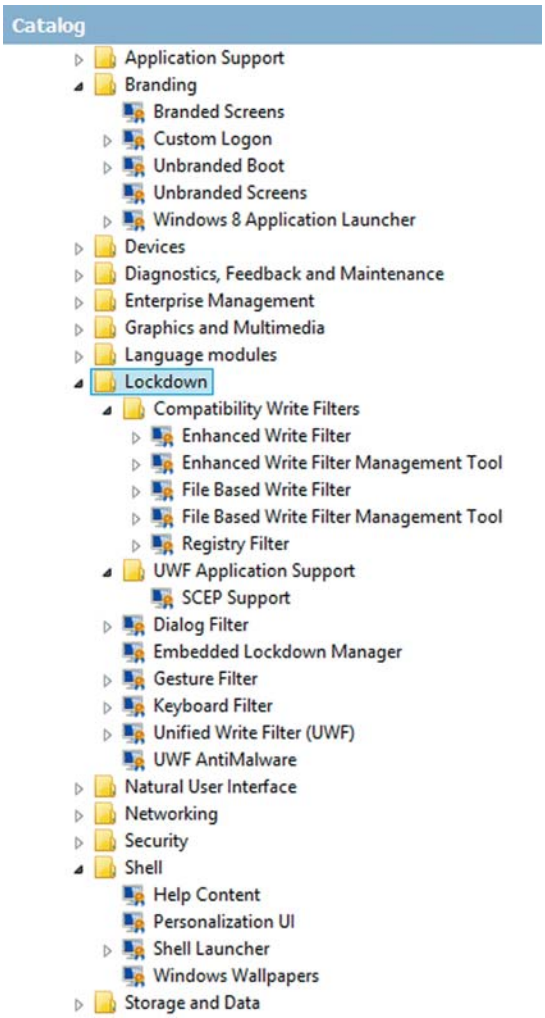

РИС. 1. Компоненты блокировки в каталоге Windows Embedded

записей на твердотельный носитель во избежание его износа, а также при работе системы на носителях только для чтения. Размер оверлея в этом случае ограничен объемом свободной памяти; при заполнении оверлея система должна быть перезагружена.

Оверлей на диске может быть гораздо большего размера, тем самым время непрерывной работы системы, связанное с заполнением оверлея, многократно возрастает. Возможность размещения оверлея на диске вместо памяти существовала также в Windows Embedded Standard 2009, но если там дисковый оверлей сохранялся после перезагрузки устройства и в любой момент можно было перенести изменения из него на диск (commit), то в Windows Embedded 8 Standard дисковый оверлей ведет себя подобно оверлею в памяти, то есть не сохраняется после перезагрузки; также в любой момент можно перенести изменения из оверлея на диск.

Если разработчиком не создано никаких исключений, то между перезагрузками оверлей не сохраняется, что позволяет получить устройство, которое после каждой перезагрузки будет находиться в исходном состоянии. Работу оверлея можно сравнить с прозрачной пленкой, наложенной на объектив проектора: любое изменение на такой пленке отражается на изображении, однако если пленку убрать, картинка останется неизменной.

Работа с защищенным носителем полностью прозрачна для приложений: с их стороны защищенный фильтром носитель ничем не отличается от обычного с возможностью записи. Все попытки записи обрабатываются фильтром автоматически. При чтении, если запрашиваемая область ранее записывалась в оверлей, возвращаются данные именно из оверлея.

Такое поведение становится интересным, когда рассматривается воздействие на систему вредоносных программ: такие программы вместе с нежелательными изменениями, внесенными ими в систему, будут существовать только до ближайшей ее перезагрузки, после которой оверлей с изменениями будет очищен и система вернется в исходное состояние.

Windows Embedded 8 Standard поддерживает следующие фильтры записи:

- EWF (Enhanced Write Filter, улучшенный фильтр записи);
- FBWF (File Based Write Filter, файловый фильтр записи);
- UWF (Unified Write Filter, объединенный фильтр записи).

UWF является новым в Windows Embedded 8 Standard и объединяет вместе возможности EWF, FBWF и фильтра реестра, поэтому нельзя одновременно использовать UWF и любой из перечисленных фильтров.

Чтобы понять различия фильтров записи, рассмотрим таблицу 1.

Различия между фильтрами объясняются тем, что EWF работает на секторном уровне, а FBWF — поверх файловой системы, что и позволяет настраивать указанный фильтр на уровне отдельных файлов. UWF объединяет достоинства обоих фильтров: он работает на секторном уровне, а также позволяет настроить фильтрацию на уровне отдельных

ТАБЛИЦА 1. РАЗЛИЧИЯ ФИЛЬТРОВ ЗАПИСИ

Функционал фильтра	UWF	EWF	FBWF
Исключения: директории и файлы	да	нет	да
Исключения: реестр	да	с использованием Registry Filter	с использованием Registry Filter
Фильтр на уровне секторов	да	да	нет
Поддержка HORM	да	нет	нет
Оверлей в памяти	да	да	да
Оверлей на диске	да	нет	нет
Провайдеры Windows Management Instrumentation v2	да	нет	нет
Сохранение раздела из оверлея на носитель	нет	да	нет
Сохранение файла из оверлея на носитель	да	нет	да
Конфигурирование на работающей системе (runtime)	командная строка, PowerShell, Embedded Lockdown Manager, провайдеры WMI	командная строка	командная строка

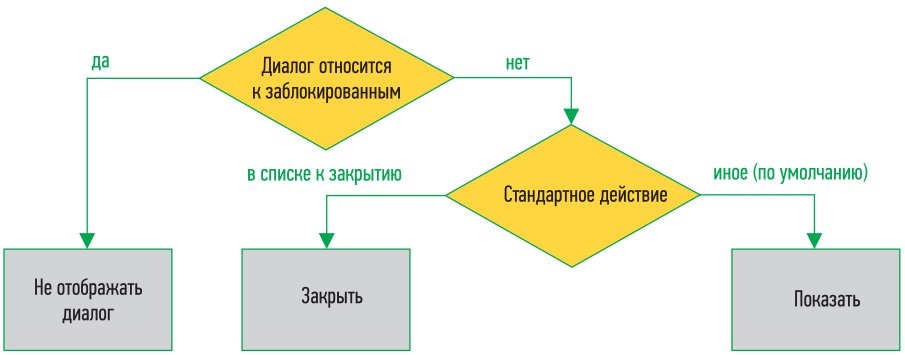

РИС. 2. Поведение фильтра диалоговых окон

файлов. Фактически фильтры EWF и FBWF оставлены разработчиками только для обратной совместимости с Windows Embedded Standard 7.

С фильтром UWF связана возможность многократно восстанавливать систему из единожды инициированного спящего режима (HORM). Это достигается путем повторного использования файла дампа памяти спящего режима после перезагрузки (в отличие от классической Windows, где указанный файл используется лишь единожды). Использование HORM несовместимо с любыми исключениями фильтров, а также с размещением оверлея на диске.

Отдельного обсуждения требует установка обновлений на систему, защищенную фильтрами записи: если не изменить поведение фильтров записи, то все изменения будут потеряны после перезагрузки устройства. Чтобы этого не произошло, для фильтров FBWF и EWF предусмотрен следующий сценарий:

1. Выключить фильтр и перезагрузить систему для вступления изменения в силу.
2. Установить необходимые обновления, при необходимости перезагрузить систему.
3. Включить фильтр и перезагрузить систему для вступления изменения в силу.

Для установки обновлений на систему, защищенную UWF, предусмотрен специальный режим обслуживания (servicing). Все действия выполняются автоматически специальным сценарием, участие администратора не требуется. Для входа в указанный режим необходимо выполнить одну команду и перезагрузить устройство. Кроме того, существует специальный компонент UWF Anti-Malware, позволяющий автоматически добавить в исключения фильтра UWF конфигурацию обновлений «Защитника Windows» (Windows Defender) так, что они будут сохраняться после перезагрузки системы.

Для развертывания (deploy) образа системы, включающей фильтры записи, перед захватом (capture) образа фильтр следует отключить. Включение фильтров можно произвести на вновь разворачиваемой системе как вручную, так и автоматически после развертывания.

ФИЛЬТР РЕЕСТРА

Фильтр реестра позволяет сохранить измененные значения отдельных разделов или параметров реестра между перезагрузками, в то время как носитель, на котором расположены файлы данных реестра, защищен фильтром записи.

Обычно в системе, защищенной фильтром записи, все изменения, производимые в реестре, попадают в оверлей и остаются там до перезагрузки. Фильтр реестра отслеживает обновления отдельных разделов или параметров и сохраняет их в свой собственный оверлей. После перезагрузки устройства изменения, сохраненные в оверлее, копируются в память, чтобы создать эффект сохранения настроек. Фильтр реестра, скомбинированный с EWF или FBWF, позволяет сохранять значения разделов или параметров реестра в то время, как оставшаяся часть системы остается защищенной фильтрами записи.

Подчеркнем следующее:

- Фильтр реестра не применяется совместно с фильтром UWF, так как последний имеет свои собственные возможности, связанные с реестром (см. таблицу 1).
- Как видно из описания, фильтр реестра, в отличие от фильтров записи, позволяет именно сохранять изменения в реестре между перезагрузками, а не уничтожать их, то есть фактически позволяет создать исключения, связанные с реестром, для фильтров EWF и FBWF.

ИСКЛЮЧЕНИЯ ФИЛЬТРОВ ЗАПИСИ

Некоторые фильтры записи допускают настройку исключений. Так, например, при использовании FBWF можно исключить из фильтра определенные файлы. В то время как указанные файлы будут записываться на защищаемый носитель, оставшаяся его часть будет по-прежнему защищена от записи.

В исключения, например, рекомендуется вносить файлы и параметры реестра, связанные с CEIP (Customer Experience Improvement Program, программа улучшения качества обслуживания) и настройками сети.

ФИЛЬТР ДИАЛОГОВЫХ ОКОН

Фильтр диалоговых окон используется для управления окнами, отображаемыми на экране, путем совершения выбранного заранее одного из действий: блокирование или выполнение стандартного действия. Блокируются все диалоговые окна, которые соответствуют заранее заданному списку правил. Среди таких правил, например, находятся заголовок окна, путь к процессу, создавшему окно, имена и типы компонентов верхнего уровня, относящихся к окну.

Для незаблокированных диалоговых окон задается стандартное действие: показать или закрыть окно (по умолчанию оно отображается). Существует также возможность всегда отображать незаблокированные окна определенных («защищенных») процессов вне зависимости от выбранного стандартного действия. Подробнее поведение фильтра показано на рис. 2.

Фильтр диалоговых окон имеет два важных ограничения:

- Он не может блокировать диалоговые окна, созданные приложениями, выполняющимися от имени администратора. На реальной системе процессы, взаимодействующие с пользователем, обычно выполняются с ограниченными правами, поэтому эта особен-

ность не сказывается на удобстве использования фильтра.
- Он анализирует только диалоговые окна, имеющие окно рабочего стола в качестве родительского. Это предотвращает затрагивание фильтром элементов, имеющих другие родительские окна (например, кнопки).

ФИЛЬТР КЛАВИАТУРЫ

Фильтр клавиатуры используется для блокирования нежелательных нажатий клавиш или их комбинаций. Обычно пользователь может использовать некоторые служебные комбинации клавиш, такие как <Ctrl+Alt+Delete>, тем самым изменяя функционирование устройства, например блокируя экран или используя диспетчер задач для закрытия приложения. Фильтр клавиатуры позволяет подавить любые нажатия клавиш или их комбинаций, приводящие к такому нежелательному поведению системы.

В Windows Embedded 8 Standard данный фильтр одинаково хорошо работает как с физическими, так и с экранными клавиатурами. Корректно отслеживаются переключения раскладки, даже если размещение подавляемых клавиш при этом изменилось.

Фильтр позволяет подавить комбинации клавиш, даже если их источником являются несколько разных клавиатур; может быть отдельно включен или выключен для учетных записей администраторов; позволяет задать правила блокирования как для скан-кодов клавиатуры, так и для виртуальных клавиш.

ФИЛЬТР ЖЕСТОВ

В настольной Windows 8 широко применяются жесты. Но во встраиваемых системах их использование может быть нежелательно, поскольку, например, пользователь может выйти на экран «Пуск», используя жест в правой части дисплея.

Фильтр жестов позволяет полностью выключить жесты на любом из краев экрана, как выборочно, в любой комбинации, так и все сразу. Обрабатываются как жесты пальцами, так и указателем мыши.

Настройка фильтра жестов производится до развертывания системы, но существуют также недокументированные возможности его настройки на работающей системе. Один из подобных способов описан в [1].

БРЕНДИРОВАНИЕ

Под брендированием подразумевается возможность изменения в системе визуальных элементов, позволяющих ее идентифицировать (например, флажок Windows при загрузке) и добавление своих собственных. Windows Embedded 8 Standard включает несколько модулей, позволяющих настраивать элементы брендирования.

Загрузка без элементов брендирования (Unbranded Boot) позволяет удалить во время загрузки элементы интерфейса, идентифицирующие систему как Windows Embedded 8 Standard, а также подавить появление экрана с ошибкой, после которой система не сможет восстановиться. Unbranded Boot настраивается как до развертывания системы, так и с командной строки на работающей системе. Существуют следующие ограничения:
- Для инициализации параметров Unbranded Boot в реестре первый вход в развернутую систему необходимо выполнить под учетной записью с административными правами.
- При использовании Unbranded Boot недопустимо конфигурировать автоматический вход в систему (Auto Logon) до развертывания образа. Необходимо делать это только на развернутой системе после первого входа с учетной записью администратора.
- Unbranded Boot не может убрать или изменить загрузочный логотип BIOS, так как он отображается до загрузки операционной системы. Однако существуют способы сделать это, если целевое устройство поддерживает UEFI (Unified Extensible Firmware Interface).

По аналогии с Unbranded Boot, собственный вход в систему (Custom Logon) позволяет избавиться от элементов брендирования уже не при загрузке, а на экране входа в систему и выключения устройства. По отдельности можно убрать, например, такие элементы экрана входа, как анимацию, кнопку выключения, выбор метода ввода, окно закрытия приложений при выключении и т. д. В качестве дополнительных возможностей в качестве опции для Custom Logon настраивается автоматический вход в систему. Подробнее про Auto Logon можно прочитать в [2].

Запуск оболочки (Shell Launcher) позволяет заменить оболочку со стандартного проводника на любое приложение, причем по отдельности указать его для различных групп или пользователей,

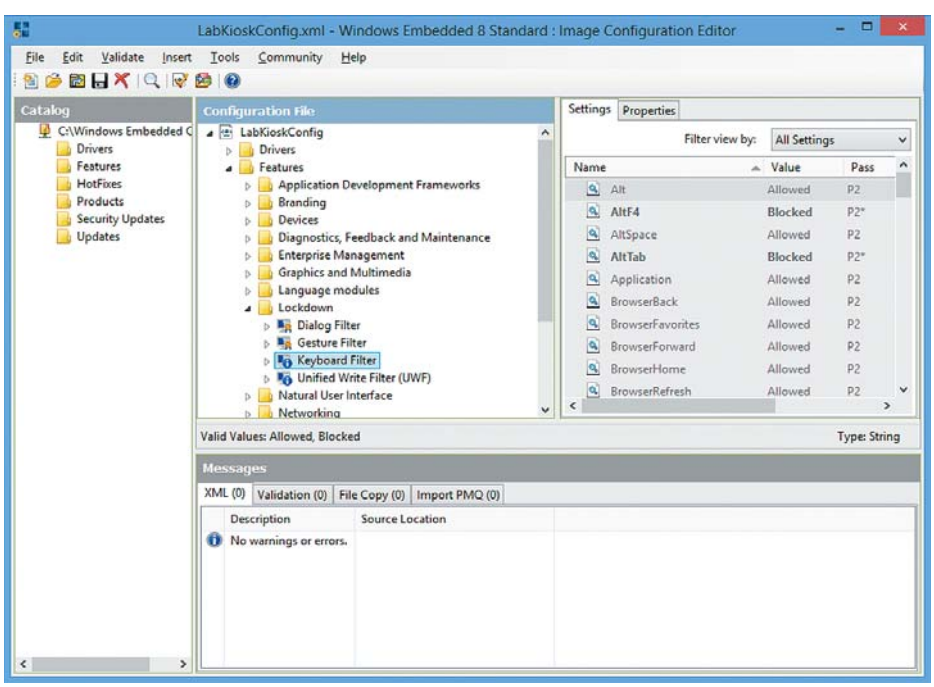

РИС. 3. ▼ Управление возможностями блокировки в редакторе конфигурации образа

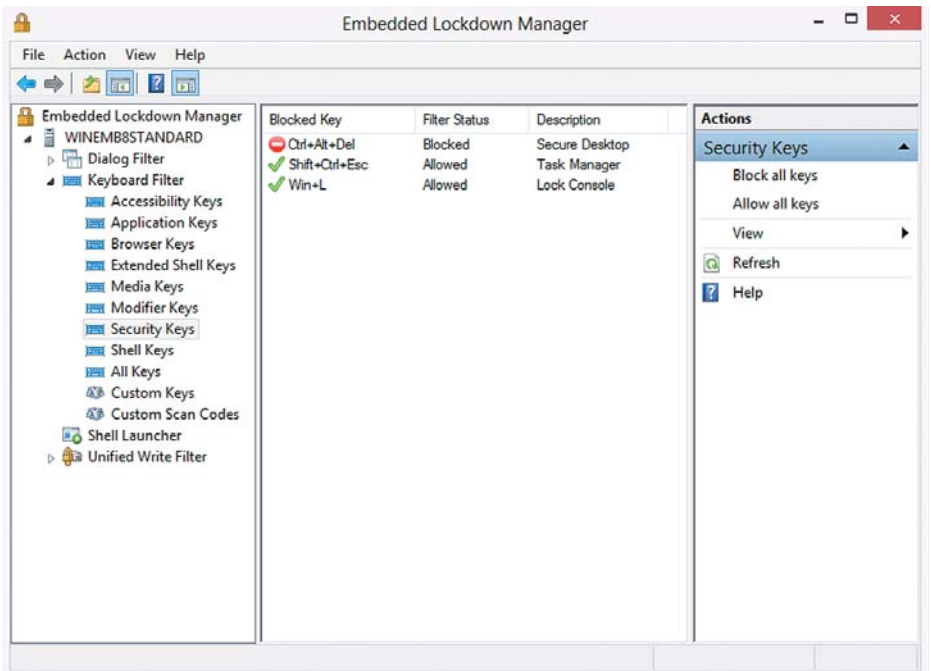

РИС. 4. Управление возможностями блокировки в менеджере блокировки

а также указать действие, которое выполняется при закрытии оболочки, например перезапуск устройства, перезапуск оболочки и т. д. Подробнее про компонент можно прочитать в [3].

Запуск приложения Windows 8 (Windows 8 Application Launcher), в отличие от запуска оболочки, позволяет автоматически запустить не классическое приложение, а приложение Windows 8 с Modern-интерфейсом после входа в систему. Настройка Application Launcher несколько отличается от Shell Launcher, так как приложения идентифицируются не путем к исполняемому файлу, а идентификатором специального вида, который носит название AUMID (Application User Model ID). Кроме того, Modern-приложения по своему поведению и техническим особенностям отличаются от классических приложений. AUMID установленных приложений можно узнать, например, с помощью командлетов PowerShell.

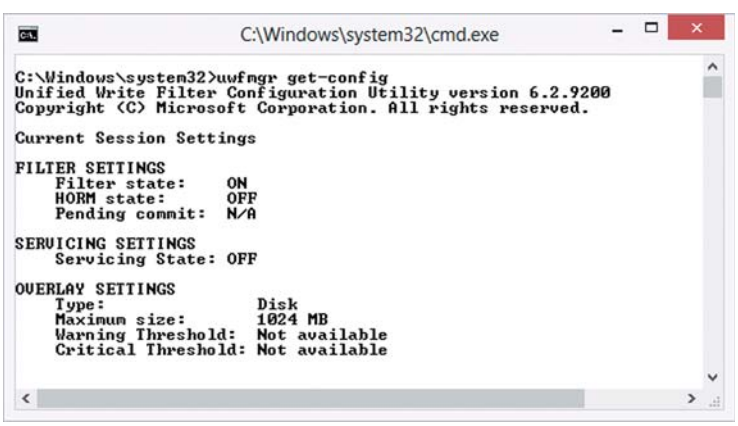

РИС. 5. Пример использования командной строки для получения текущей конфигурации UWF

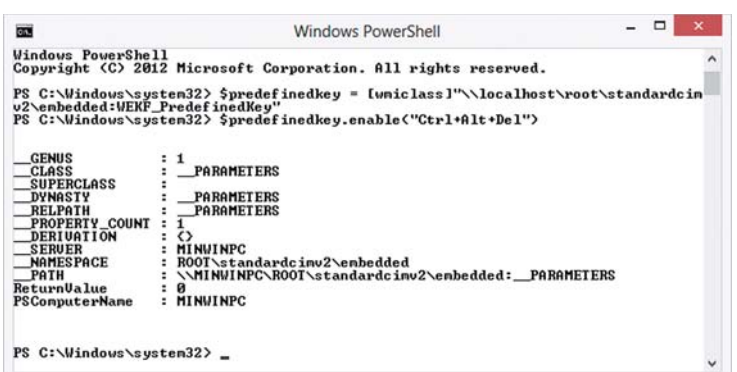

РИС. 6. Пример использования PowerShell для включения комбинации клавиш фильтра клавиатуры

УПРАВЛЕНИЕ ВОЗМОЖНОСТЯМИ БЛОКИРОВКИ

Возможности блокировки могут быть настроены в ICE (Image Configuration Editor, редактор конфигурации образа) на этапе проектирования образа (рис. 3), непосредственно на работающей системе (различными способами), а также с помощью инструмента ELM (Embedded Lockdown Manager, менеджер блокировки).

ELM позволяет производить настройку не только локально, непосредственно на целевой машине, но и удаленно, по сети. Для этого необходимо использовать учетную запись администратора с установленным не пустым паролем. Кроме того, необходимо сделать ряд настроек на целевой системе, чтобы разрешить удаленное управление возможностями изоляции.

Установка ELM для удаленного управления производится запуском на компьютере разработчика одного из файлов *Windows8-RT-KB2758707-x86.msu* или *Windows8-RT-KB2758707-x64.msu* (в зависимости от разрядности системы) с дистрибутива Windows Embedded 8 Standard Toolkit или по ссылке [4]. Для установки ELM на целевой системе не следует запускать указанные файлы, следует включить ELM в образ системы на этапе его разработки или развертывания.

В ELM (рис. 4) отображаются только доступные на целевой машине возможности изоляции. Те возможности, которые не были включены в образ, не будут доступны. ELM поддерживает настройку UWF, Dialog Filter, Keyboard Filter и Shell Launcher. Все параметры,

доступные для редактирования в ICE при создании файла ответов, доступны также для редактирования в ELM во время выполнения целевой системы.

Среди других путей управления блокировкой также присутствует использование командной строки (рис. 5) или командлетов PowerShell (рис. 6). Не все возможности блокировки управляются всеми перечисленными способами, для детальной информации следует обратиться к документации, прилагаемой к ICE, или найти ее по ссылке [5].

Общая информация о системе Windows Embedded 8 Standard может быть найдена по ссылкам [6, 7].

ВЫВОДЫ

Возможности блокировки являются ключевым отличием Windows Embedded от классических систем Windows и позволяют добиться желаемого поведения системы, не тратя время и силы на ее адаптацию ко встраиваемому применению.

В следующей статье цикла речь пойдет о редакторе компонентов Module Designer, позволяющем создавать собственные компоненты каталога Windows Embedded для последующего встраивания их в образ системы. ⬣

ЛИТЕРАТУРА:

1. http://msembedded.ru/?p=2553
2. http://msembedded.ru/?p=2513
3. http://msembedded.ru/?p=2579
4. http://www.microsoft.com/en-us/download/details.aspx?id=37020
5. http://msdn.microsoft.com/en-US/library/ff795586(v=winembedded.0).aspx
6. http://www.getwindowsembedded8.com
7. http://www.microsoft.com/embedded

Компания «Кварта Технологии», основанная в 1997 г., является одним из лидеров российского ИТ-рынка в области дистрибуции и продажи программных продуктов. Являясь дистрибутором и тренинг-партнером корпорации Microsoft в области встраиваемых решений, компания осуществляет поставку лицензий и средств разработки, предоставляет полную информационную и техническую поддержку, услуги по разработке образов под нужды заказчика, консалтинг, обучение специалистов и проведение сертифицированных тренингов по встраиваемым технологиям Microsoft.

На ежегодной конференции «Встраиваемые технологии 2013. Современные программные и аппаратные решения» в апреле 2013 г. «Кварта Технологии» и ее партнеры представили готовые решения на базе Microsoft Windows Embedded. В этом году состоялся запуск сразу нескольких новых продуктов: Windows Embedded 8 Standard, Windows Embedded 8 Professional, Windows Embedded 8 Industry, а также была анонсирована новая ОС Windows Embedded Compact 2013, которая планируется к выходу в сентябре 2013 г.

По данным «Кварта Технологии», наиболее полно решения Microsoft Windows Embedded используются в таких отраслях, как разработка и производство компьютерного оборудования и комплексных решений, а также разработка программного обеспечения. Если посмотреть на статистику по типам решений, то лидируют системы автоматизации предприятий, программное обеспечение для встраиваемых устройств, платежные и информационные киоски, промышленные контроллеры, серверы, тонкие клиенты, измерительные устройства.