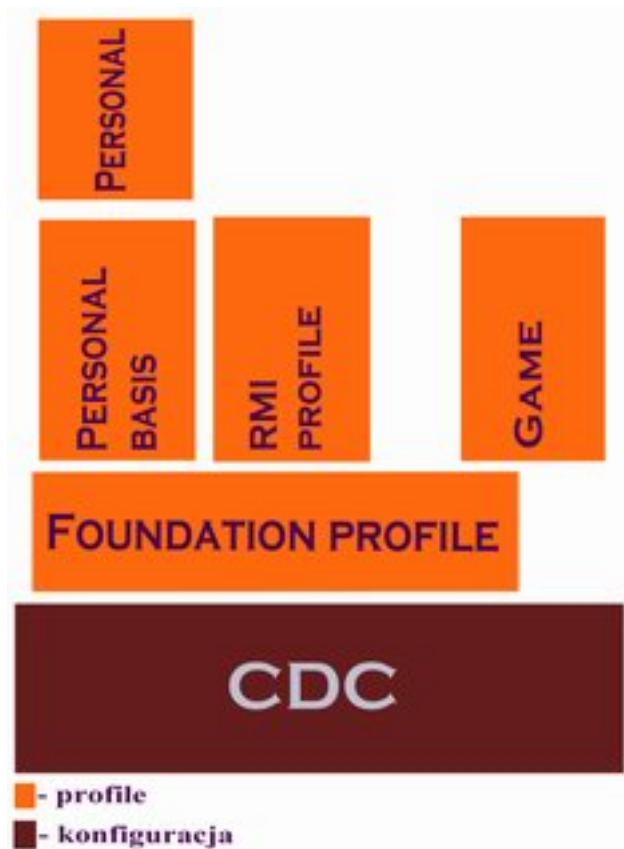


CDC - widok ogólny

Konfiguracja CDC ma za zadanie stworzyć podstawy dla profili przeznaczonych dla urządzeń pomiędzy słabymi telefonami czy PDA, a zwykłymi PC- tami. Są to więc urządzenia już dużo mocniejsze od tych z klasy urządzeń CLDC. Najczęściej posiadają już 32-bitowy procesor (telefony - 16-bitowy) i dużo bardziej zaawansowane połączenie sieciowe.

Inna jest również architektura profili opartych na konfiguracji CDC. Przedstawia ją poniższy szkic:



Konfiguracja CDC i jej profile.

Na konfiguracji CDC oparty jest profil Foundation Profile (specyfikacja dostępna na stronie: <http://jcp.org/jsr/detail/46.jsp>). Rozszerza on omawianą konfigurację o klasy pakietów już zawartych w CDC, jak i poszerza ją o nowe pakiety. Główną funkcjonalnością, którą ten profil się nadal *nie* zajmuje jest interfejs użytkownika.

Na tym profilu oparte są kolejne:

1. Profile: *Personal Basis* i *Personal*. Dodają one interfejs użytkownika do profilu Foundation. Planowane jest zastąpienie nimi używanego obecnie Personal Java. Szczegóły na temat stanu rozwoju tych profili znaleźć można pod adresami:
 1. <http://jcp.org/jsr/detail/129.jsp> - Personal Basis
 2. <http://jcp.org/jsr/detail/62.jsp> - Personal
2. *Profil RMI* dodaje możliwość zdalnego wywoływania metod (od technologii RMI).

Oczywiście zaimplementowano jedynie stronę klienta, jako że urządzenia przenośne pełnią tylko taką rolę.

Profil RMI rozwijany jest jako JSR o numerze 66.

3. *Game Profile* jest profilem pomyślanym jako rozszerzenie CDC dla deweloperów gier. Nie jest jeszcze jasne, czy profil ten oparty będzie na profilu Foundation, czy bezpośrednio na konfiguracji CDC.

Szczegóły pod adresem: <http://jcp.org/jsr/detail/134.jsp>

Zawartość CDC

Napisałem, że urządzenia, dla których projektowana była konfiguracja CDC są już urządzeniami mocnymi. Ma to oczywiście swoje przełożenie w zawartości w/w konfiguracji. Oto pakiety, których klasy znalazły się w konfiguracji CDC i różnice pomiędzy nimi a ich odpowiednikami w J2SE:

TABELA2:

Pakiet	Opis
java.io	Prawie cały pakiet został dołączony do CDC. Wyjątek stanowią rzadko używane klasy Reader i Writer oraz LineNumberInputStream i StringBufferInputStream. Dwie ostatnie nie zostały dołączone, jako nie zalecane (ang. <i>deprecated</i>) w J2SE.
java.lang	Nie dołączono jedynie klas Compiler i UnknownException
java.lang.ref	Dołączony bez zmian.
java.lang.reflect	Dołączony bez zmian.
java.math	Z dwóch klas tego pakietu dołączona została BigInteger, nie dołączono BigDecimal
java.net	CDC oferuje połączenia oparte na datagramach, ale nie udostępnia socketów i protokołu HTTP. Z tego pakietu nie zostały więc dołączone klasy oferujące te dwa rodzaje połączeń.
java.security	
java.security.cert	Zawiera jedynie klasę Certificate i dwa wyjątki związane z nią.
java.text	Pakiet zawiera klasy dot. wsparcia dla

	formatowania zależnego od lokalizacji, przetwarzanie dat i liczb oraz formatowanie komunikatów o błędach.
<code>java.util</code>	Prawie cały pakiet został dołączony. Jedyne różnice to klasy dotyczące obsługi zdarzeń (jak <code>Observer</code> czy <code>EventObject</code>) i timer'ów.
<code>java.util.jar</code>	Pakiet jest pełny, oprócz klasy <code>JarOutputStream</code> . Oznacza to brak możliwości utworzenia pliku JAR. Jest to oczywiście w przypadku CDC niepotrzebne.
<code>java.util.zip</code>	Pakiet zawiera klasy umożliwiające czytanie z plików ZIP, ale nie udostępnia strumieni umożliwiających aplikacjom zapisywanie czy kompresję i dekompresję plików ZIP.
<code>java.microedition.io</code>	Pakiet dodany dla kompatybilności z aplikacjami napisanymi dla CLDC.

W przeciwieństwie do konfiguracji CLDC, jeśli klasa została dołączona do CDC, to nie różni się ona od swojego odpowiednika w J2SE (z wyjątkiem dla metod nie zalecanych).

Maszyna wirtualna konfiguracji CDC

Dzięki odpowiednim zasobom urządzeń, na których uruchamiana jest maszyna wirtualna konfiguracji CDC, może ona być PEŁNĄ maszyną wirtualną, czyli zgodną ze specyfikacją JVM, przystosowaną jednak do ograniczonej ilości pamięci.

Do funkcjonalności KVM dodaje ona więc obsługę następujących cech języka Java dostępnych w JVM:

1. obsługa działań na liczbach zmiennoprzecinkowych,
2. uruchamianie kodu natywnego dzięki JNI,
3. obsługa słabych referencji,
4. Reflection,
5. serializację obiektów,
6. definiowane przez użytkownika class-loader'y,
7. obsługa Java Virtual Machine Debug Interface (JVMDI).

Maszyna wirtualna konfiguracji CDC nazywa się *CVM*. Początkowo był to skrót od *Compact VM*. Ale KVM jest również kompaktową, więc CVM jest teraz po prostu maszyną wirtualną konfiguracji CDC, bez rozwinięcia skrótu.

Przykładową implementację maszyny wirtualnej CVM można ściągnąć ze strony Sun'a pod adresem <http://java.sun.com/products/cdc>

Szczegóły specyfikacji konfiguracji CDC znajdziesz pod adresem: <http://jcp.org/jsr/detail/36.jsp>.