

Make Applications More Valuable



Jalapeño: pekelně ostrá Java persistence v Caché

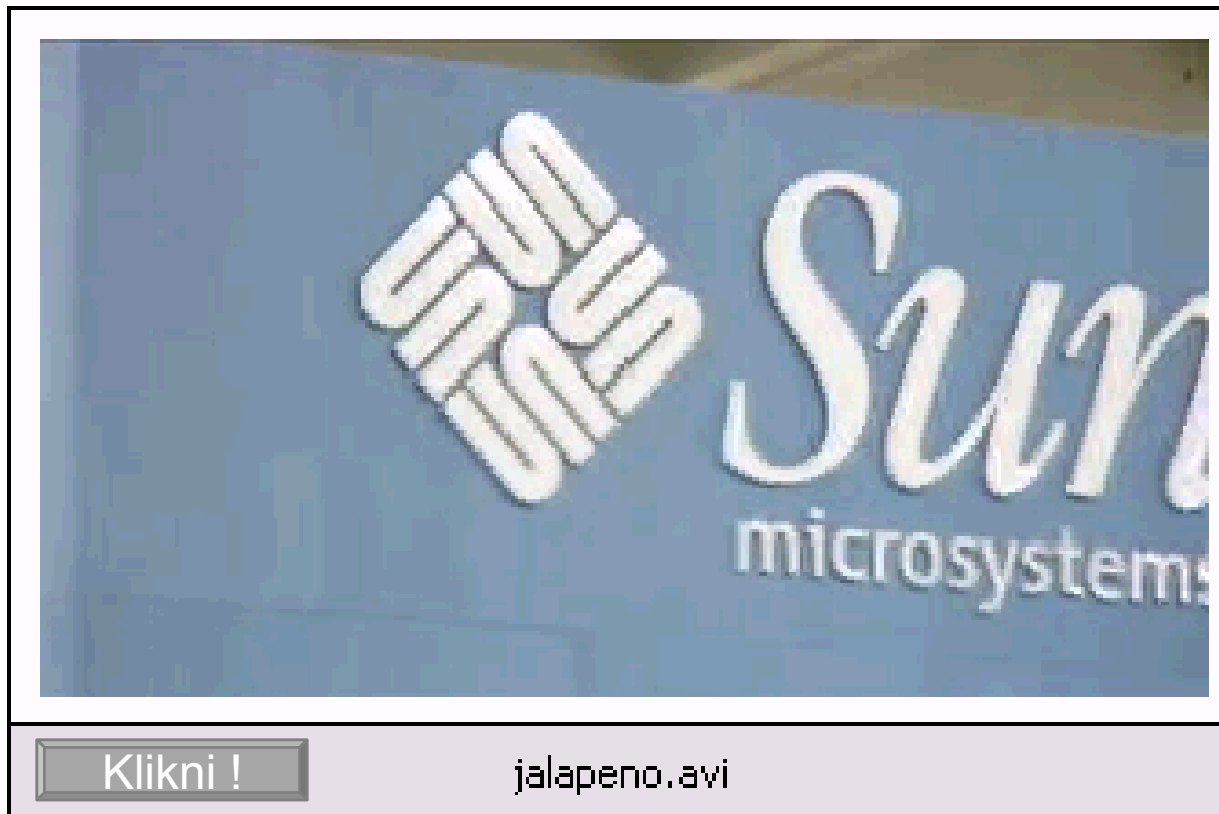
Daniel Kutáč
Senior Sales Engineer

Innovations by InterSystems

Co je Jalapeño



Pár slov ředitele vývoje software...



Tak tedy, o čem dnes budeme mluvit ...



- **Architektura**
- **Instalace**
- **Anotace**
- **Tvorba persistentních schémat**
- **API**
- **Caché: Databáze a aplikační server v jednom = RAD**

Jalapeño - Architektura



- Jalapeno Runtime Library
 - Sada Java knihoven pro připojení ke Caché a ukládání a následné dotazování nad objekty uloženými v databázi
- Jalapeno Schema Builder
 - Nástroj analyzující definice Java tříd a vytvářející odpovídající třídy Caché

Architektura - klasická



GUI

- Java třídy pro GUI rozhraní

Aplikační server

- Java třídy s aplikační logikou
- EJB server

Mapovací vrstva

- Převod mezi Java třídami a SQL tabulkami

Databáze

- SQL tabulky
- Uložené procedury

Jalapeño - Architektura



GUI

- Java třídy pro GUI rozhraní

Aplikační server

- Java třídy s aplikační logikou
- Java třídy datového modelu
- Anotace

Databáze

- Persistentní objekty

Jalapeño - Instalace



- Jalapeno je integrální součástí Caché, k dispozici od verze 5.2
- Pro některá Java IDE existují pluginy, které je možno doinstalovat pro pohodlnou práci s komponentou Schema Builder
 - NetBeans
 - Eclipse*
 - IDEA

*od 2. pol. 2007

Jalapeño - Anotace



- Jalapeno se používá nad klasickými POJO třídami nebo nad Beany
- Jalapeno se nepoužívá nad EJB, k tomu slouží jiné třídy
- Jalapeno potřebuje JVM 1.5 nebo J2SE 5

Jalapeño - Anotace



- Anotace na úrovni třídy
 - @Access
 - Level
 - Type
 - @CacheClass
 - @ClassParameter, @ClassParameters
 - @Embeddable
 - @Extends
 - @Implements
 - @Index, @Indices
 - @Transient

Jalapeño - Anotace



- Anotace na úrovni vlastností
 - @CacheProperty
 - @Collection
 - @Relationship
 - @ID
 - @Lob
 - @PropertyParameter, @PropertyParameters
 - @Transient
 - @Version

Tvorba persistentních schémat



- Caché SchemaBuilder Wizard
 - Java GUI aplikace, spustitelná z příkazové řádky
 - `com.jalapeno.tools.SchemaBuilder.class`
- Pluginy IDE
 - Příkazy menu IDE
 - Jalapeno -> Schema Management -> Build Schema

Jalapeño - API



- Vytvořením schémat vzniknou v Caché persistentní třídy
- Caché třídy obsahují vlastnosti dané anotací `@Access` plus XML kontejner obsahující některé informace používané třídou `Java ObjectManager` za běhu (např. způsob dovážení záznamů – `fetch`)



- Třída `ApplicationContext`
 - Slouží k navázání spojení s Caché prostřednictvím objektově orientovaného protokolu postaveného nad JDBC

```
Class.forName ("com.intersys.jdbc.CacheDriver");  
Connection conn = DriverManager.getConnection  
 (url, username, pwd);  
objManager =  
ApplicationContext.createObjectManager (conn);
```

Jalapeño - API



- Interface ObjectManager
 - Poskytuje funkčnost objektům manipulujícími daty v Java třídách v Caché databázi
 - Pár metod na ukázkou:
 - openById(), openByPrimaryKey(), openByQuery()
 - insert(), save(), update()
 - attach(), detach(),
 - refresh(), swizzle()
 - startTransaction(), commit()

Jalapeño - API



- Utility:
 - Interface ExtentManager
 - Poskytuje rozhraní pro práci na úrovni rozsahu Caché tříd
 - Třída Settings
 - Poskytuje rozhraní pro nastavení vlastností instance ObjectManager
 - Interface Utilities
 - Poskytuje rozhraní pro XML serializaci a deserializaci tříd

Ukázka



```
Class.forName("com.intersys.jdbc.CacheDriver");
Connection connection = DriverManager.getConnection (
    "jdbc:Cache://127.0.0.1:56773/USER", username, password);
ObjectManager om = ApplicationContext.createObjectManager (connection);

Kniha kniha = new Kniha ();
kniha.nazev = "Medvídek Pů"; kniha.autor = "A. A. Milne";
om.save(kniha, true);

Stranka prvni = new Stranka();
prvni.obsah = "Přesně to fakt už nevím, ale jedná se tu o medvědovi" +
    " a dalších zvířatech ze stokorcového lesa.";
prvni.kniha = kniha; om.save(prvni, true);

Stranka posledni = new Stranka();
posledni.obsah = "No, a to je tak zhruba všechno.";
posledni.kniha = kniha; om.save(posledni, true);

// need to refresh client after server save
om.refresh(kniha);

Utilities ut = om.utilities();

/*          zobrazeni knihy          */
Kniha k = null;
Iterator knihy = om.openByQuery(Kniha.class, "autor like ?",
    new Object[] { "%Milne%" });
while (knihy.hasNext()) {
    k = (Kniha)knihy.next();
    System.out.println(k.nazev + " počet stran: " + k.stranky.size());
    System.out.println("xml: " + ut.serializeToXML(k));
}

om.close();
```

Make Applications More Valuable

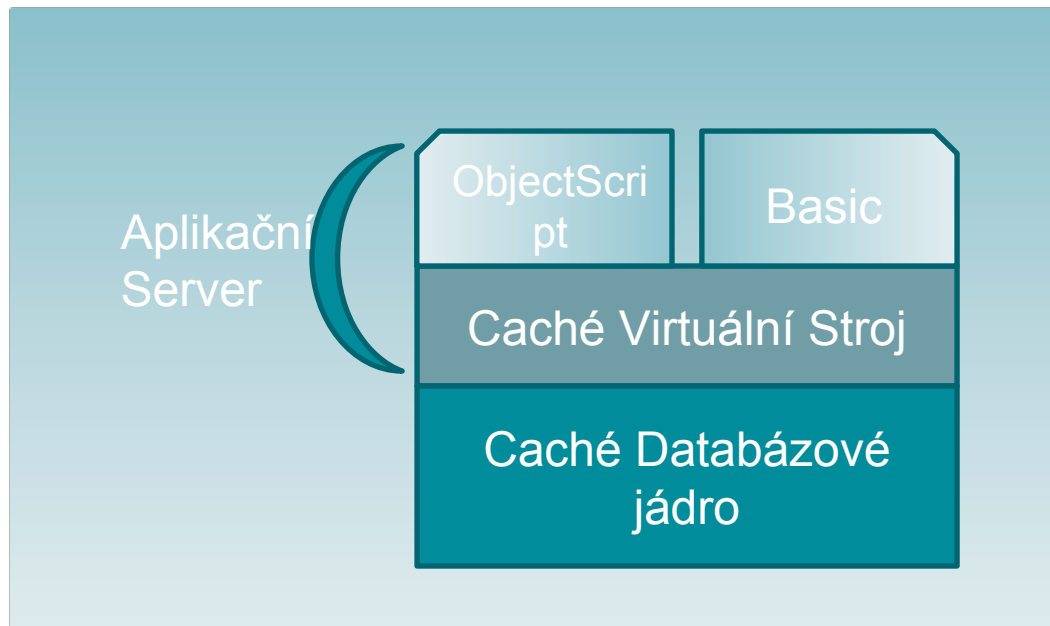


Caché

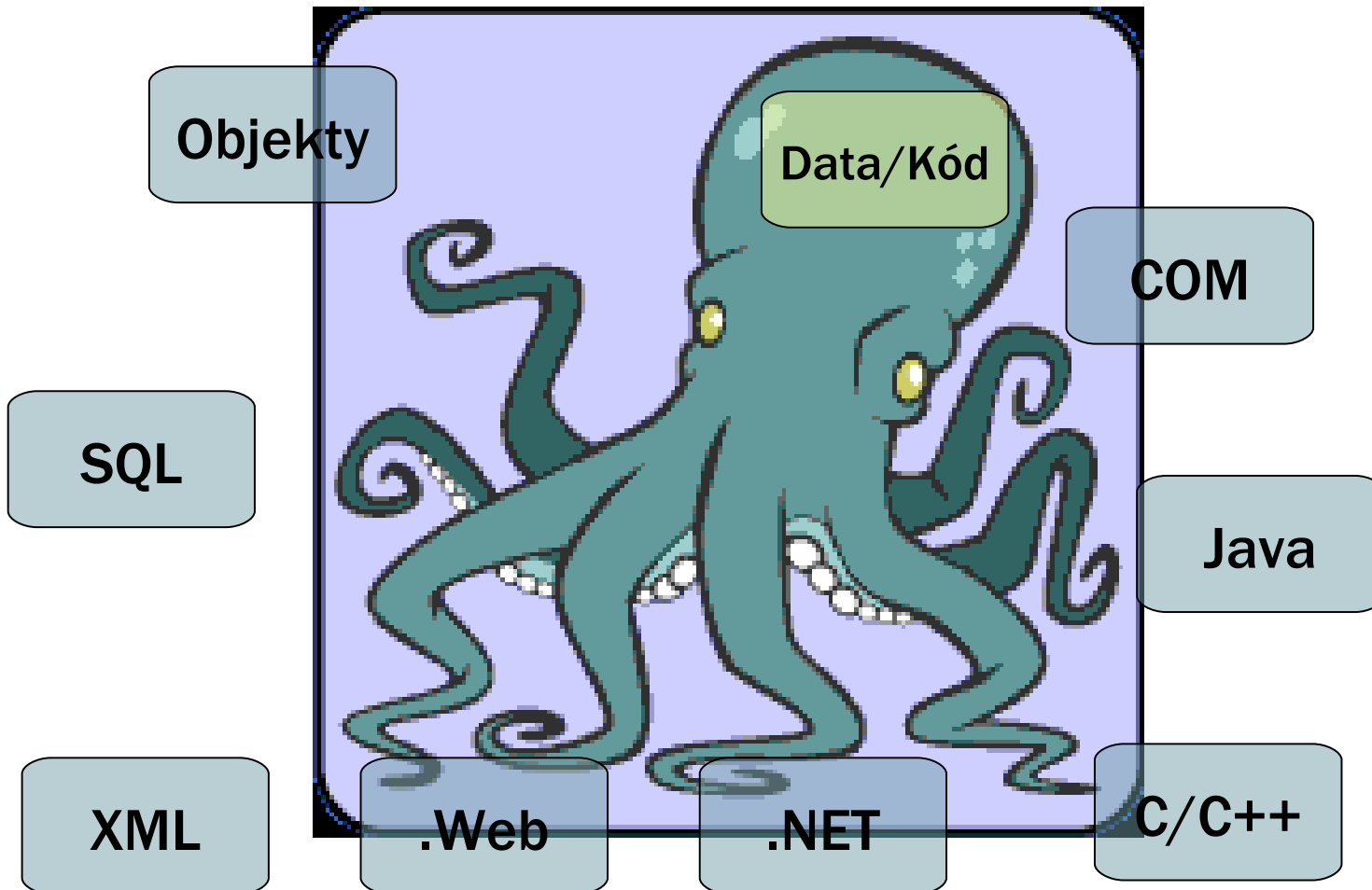
Kam že se to ty Java objekty ukládají?

Innovations by **InterSystems**

Unifikovaná Architektura Dat



Otevřený design

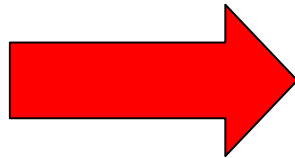


Objekty Caché



Package::class
Attributes...
Methods...

OMG objekt



Package::class
Attributes...
Methods...
<i>SQL Queries...</i>
<i>XML...</i>
<i>Indices...</i>
<i>FK...</i>
<i>Projections (.NET, Java, C/C++)...</i>
....

Caché objekt

Fyzické uložení dat v Caché a jejich získávání - 1



```
/// Common Form - defines persistency
Class Common.Forms.Form Extends (%Persistent, Common.Forms.IForm, %XML.Adaptor) [ ClassType = persistent,
{

Index IndexOnDeprecated On Deprecated [ Type = bitmap ];

Index IndexOnFormId On FormId;

/// Form sections (at least one must exist)
Relationship Sections As Common.Forms.Section [ Cardinality = children, Inverse = Form ];

Method Task()
{

write "Form: ", $c(9), ..Description, !
write $c(9), "Form identifier: ", ..FormId, !
write $c(9), "Is Deprecated: ", $s(..Deprecated:"yes", 1:"no"), !
write " ... sections ...", !!
quit:..Sections.Count()=0
// sekce
set key=""
set section=..Sections.GetNext(.key)
```

Fyzické uložení dat v Caché a jejich získávání - 2



```
Set sc=##class(Common.Forms.Form).$fulgetindices() Quit:('sc) sc
Kill ^Common.F.FormD
Quit 1
%Load(%this,id) public { Set sc=..%IdSet(id) Quit:('sc) sc
Set sc=..%LoadData(id) Quit:('sc) sc
Quit 1 }
%LoadData(%this,id) New %sc
Set %sc=""
If $ZObjVal(,1,0,3,1)=4 Lock +(^Common.F.FormD(id)):$ZUtil(115,4) If '$Test Quit $$Error^%
If $ZObjVal(,1,0,3,1)'=4,$ZObjVal(,1,0,3,1)>1 Lock +(^Common.F.FormD(id)#"S"):$ZUtil(115,4
If '$Data(^Common.F.FormD(id)) Set $ZObjVal(,3,0,3,3)="", $ZObjVal(,4,0,3,4)="", $ZObjVal(,5
Else DO
. New %s1
. Set %sc=1
. Set %s1=$Get(^Common.F.FormD(id))
. Set $ZObjVal(,3,0,3,3)=$ListGet(%s1,2), $ZObjVal(,4,0,3,4)=$ListGet(%s1,3), $ZObjVal(,5,0,
If $ZObjVal(,1,0,3,1)=2 Lock -(^Common.F.FormD(id)#"SI")
Quit $Select(%sc'="" :%sc,1:$$Error^%apiOBJ(5809))
%New(initvalue) public { Quit ##class(Common.Forms.Form).%New($get(initvalue)) }
%Construct(%this.initvalue) Set $ZObjVal(,1,0,3,1)=$ZUtil(115,10).$ZObjVal(,2,0,3,2)=0.$ZObjVal(
```

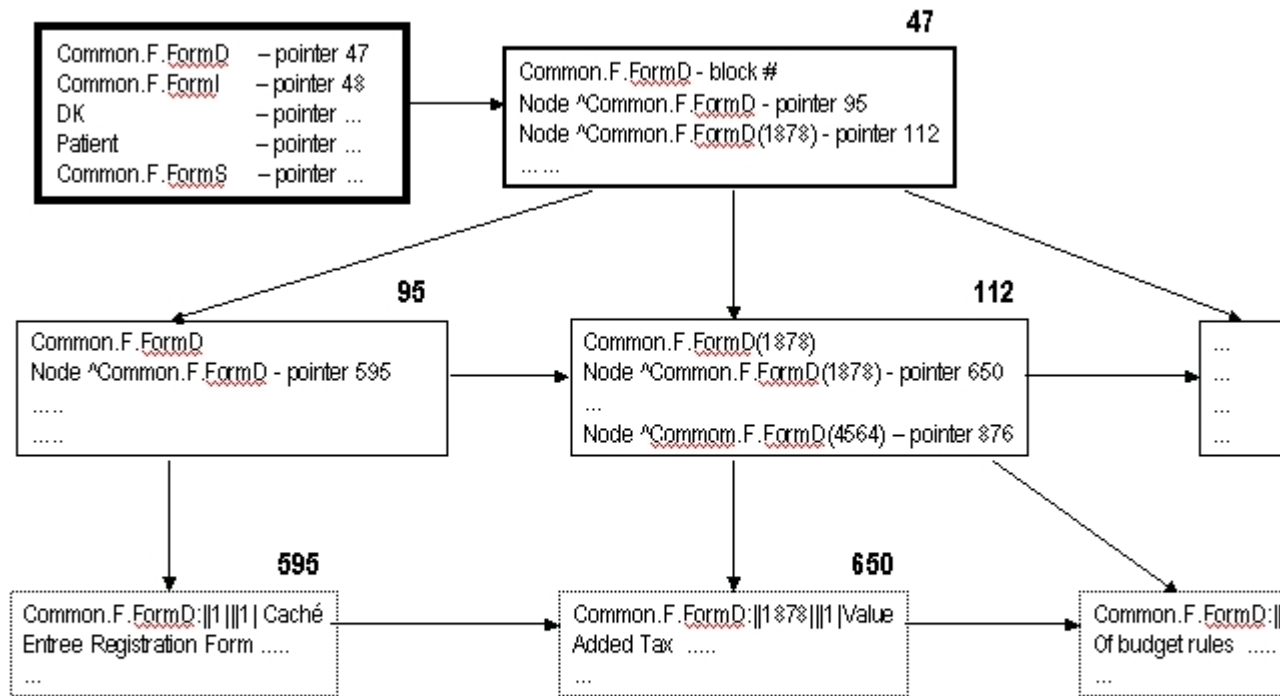
Runtime kód Caché serveru vygenerovaný na základě definic tříd

Fyzické uložení dat v Caché a jejich získávání - 3



```
^Common.F.FormD=4
^Common.F.FormD(1)=<<$LB("", "1", "Cache Entree Registration Form", "CE001", "58805", "5911")
^Common.F.FormD(1, "Sections", 1)=<<$LB("", "Default")>>
^Common.F.FormD(1, "Sections", 1, "Elements", 1)=<<$LB("", "Name", "", "30")>>
^Common.F.FormD(1, "Sections", 1, "Elements", 2)=<<$LB("", "Date of training", "", "8")>>
^Common.F.FormD(1, "Sections", 1, "Elements", 3)=<<$LB("", "Company", "", "30")>>
^Common.F.FormD(1, "Sections", 1, "Elements", 4)=<<$LB("", "e-mail", "", "30")>>
^Common.F.FormD(2)=<<$LB("", "0", "Cache Entree Registration Form (2003)", "CE002", "59170")
^Common.F.FormD(2, "Sections", 2)=<<$LB("", "Default")>>
^Common.F.FormD(2, "Sections", 2, "Elements", 5)=<<$LB("", "Name", "", "30")>>
^Common.F.FormD(2, "Sections", 2, "Elements", 6)=<<$LB("", "Date of training", "", "8")>>
^Common.F.FormD(2, "Sections", 2, "Elements", 7)=<<$LB("", "Company", "", "30")>>
^Common.F.FormD(2, "Sections", 2, "Elements", 8)=<<$LB("", "e-mail", "", "30")>>
^Common.F.FormD(2, "Sections", 2, "Elements", 9)=<<$LB("", "Training fee payment method", "", "1")>>
^Common.F.FormD(4)=<<$LB("~Common.Forms.TaxForm~", "0", "Income Tax - physical persons")
^Common.F.FormD(4, "Sections", 3)=<<$LB("", "Header")>>
^Common.F.FormD(4, "Sections", 3, "Elements", 10)=<<$LB("", "Reg.#", "", "10")>>
^Common.F.FormD(4, "Sections", 3, "Elements", 11)=<<$LB("", "Tax.#", "", "14")>>
^Common.F.FormD(4, "Sections", 4)=<<$LB("", "Paragraph 6, income from employment")>>
^Common.F.FormD(4, "Sections", 4, "Elements", 12)=<<$LB("", "Income from employment - 1", "", "1")>>
^Common.F.FormD(4, "Sections", 4, "Elements", 13)=<<$LB("", "Income from employment - 2", "", "1")>>
^Common.F.FormD(4, "Sections", 5)=<<$LB("", "Paragraph 7, income from capital (interest, ...)")>>
^Common.F.FormD(4, "Sections", 6)=<<$LB("", "Paragraph 10, lease income")>>
```

Fyzické uložení dat v Caché a jejich získávání - 4



Legend:

directory block

top pointer block

bottom pointer block

data block

Caché není žádné ořezávání...



- Největší instalace:
 - Uživatelé: cca 35 000
 - Databáze: cca 10 TB*

- Podpora škálování a odolnosti proti výpadkům
 - Failover clustery
 - ECP pro distribuované systémy
 - Stínování pro rozložení zátěže

*Stejná data v relační DB mohou zabírat 1 – 10x více objemu

Doporučuji Vaší pozornosti



- Java Feature – Evaluating Options for Persisting Java Objects
 - <http://java.sys-con.com/read/377037.htm>
- IBM – projekt FocalPoint
- <http://www.intersystems.com/java>

Kde se dozvíte více?



Symposium InterSystems 2007

"Polepšené aplikace"

18. - 19. června 2007, Špindlerův Mlýn, hotel Horal



mediální partner

Innovations by InterSystems