

INTERSYSTEMS EDUCATION

Caché & Java – Overview

Documentação Base

Esta documentação foi baseada no Caché versão 4.1.6 para a plataforma Windows.

Mas é aplicável de maneira geral as versões do grupo 4.

Pré-requisitos recomendados

- Estar familiarizado com a definição de Classes Caché;
- Conhecimento básico em Java;
- Conhecimento básico em SQL e/ou Caché Objetos.

Softwares utilizados

- Caché 4.1.6;
- Java 2 Standard Edition 1.4.1.

Caché – Detalhes de instalação

- Caché 4.1.6 instalado em C:\CacheSys;
- Subdiretório dos componentes para Java:
 - C:\CacheSys\Java
 - CacheJava.jar
 - CacheJDBC.jar

Java – Detalhes de instalação

- Java 2 SDK, Standard Edition Version 1.4.1 instalado em:

C:\j2sdk1.4.1

Diretório de trabalho para Java

- Criar um diretório de trabalho para usar com os exemplos de Java:

C:\TrabalhoJava

Java – Variável CLASSPATH

Todo o comando em uma linha.

```
SET CLASSPATH= C:\CacheSys\Java\CacheJava.jar;  
C:\CacheSys\Java\CacheJDBC.jar; C:\TrabalhoJava;  
%CLASSPATH%
```

Java – Variável PATH

Todo o comando em uma linha.

```
SET PATH= C:\j2sdk1.4.1\bin; "%PATH%"
```

Java – Variável PATH e CLASSPATH

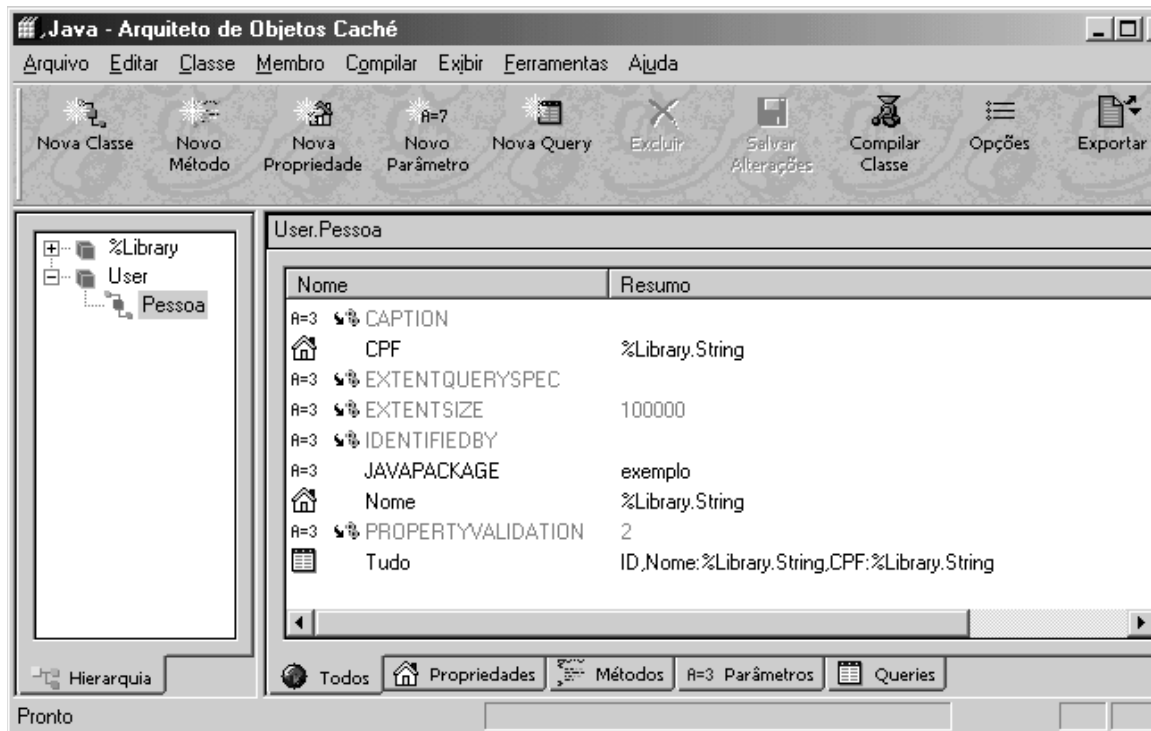
- Podem ser definidas em um arquivo de lote (.bat);
- Ou de forma permanente na inicialização do Windows.

Exemplo (C:\TrabalhoJava\setenv.bat):

```
SET CLASSPATH=C:\CacheSys\Java\CacheJava.jar;  
C:\CacheSys\Java\CacheJDBC.jar;C:\TrabalhoJava;%CLASSPATH%  
SET PATH=C:\j2sdk1.4.1\bin;"%PATH%"
```

Classe exemplo no Caché

- Criar um classe persistente para exemplo no Caché:
 - Nome da Classe: **User.Pessoa**
 - Propriedades: **Nome** e **CPF**
 - Query: Tudo
(**SELECT ID, Nome, CPF FROM Pessoa**)
 - Atributo: **JAVAPACKAGE = exemplo**
 - Namespace no Caché utilizado nos exemplos desta apresentação : **JAVA.**



- Salvar e compilar a classe no Caché;
- No Arquiteto de Objetos do Caché, selecionar a classe e em seguida selecionar a opção "Gerar Java";
- Selecionar o diretório de trabalho para a exportação da classe (**C:\TrabalhoJava**).

Classe do Caché exportada para Java

- Após a exportação, no diretório **C:\TrabalhoJava**, teremos um sub-diretório **exemplo** devido ao atributo **JAVAPACKAGE** que foi definido na classe Caché;
- E aí teremos o arquivo **Pessoa.Java** com a definição da classe **Pessoa** no formato Java.
- Prompt do MS-Dos;
- Posicionar-se no diretório **C:\TrabalhoJava**;
- Ajustar as variáveis de ambiente com **setenv.bat**;
- Posicionar-se no diretório **C:\TrabalhoJava\exemplo**;
- Compilar classes Java com o comando **javac**:
javac Pessoa.java

Programação em Java

- Duas formas de conectividade entre Caché e Java estão disponíveis:
 - Acesso nativo as classes Caché (Caché Java Binding);
 - Acesso via JDBC (SQL).

Java – Exemplos de acesso nativo

- Aplicativo java **exemplo1a** insere um novo objeto na classe **Pessoa**;
- Aplicativo java **exemplo1b** executa uma pesquisa (query) na classe **Pessoa**.

Java – exemplo1a (acesso nativo)

O aplicativo em Java do exemplo a seguir recebe dois valores por argumentos (**Nome** e **CPF**) e cria um novo objeto na classe **Pessoa**, acessando o Caché via Caché Java Binding:

```
package exemplo;
import COM.intersys.objects.*;
import COM.intersys.util.*;
class exemplo1a {
    public static final String CACHE_URL =
cn_iptcp:127.0.0.1[1972]:JAVA";
    static ObjectFactory factory = null;
    public static void main(String args[]) {
        try {
            factory = new ObjectFactory(CACHE_URL);
            Pessoa p = null;
            p = new Pessoa(factory);
            p.setNome(args[0]);
            p.setCPF(args[1]);
            p._save();
            p._close();
        }
        catch (CacheException e) {
            System.out.println("Erro no Caché: " +
e.getMessage());
            e.printStackTrace();
        }
        catch (ArrayIndexOutOfBoundsException e) {
```

Requisitos

Forma de conexão
 Endereço do servidor Caché
 Porta TCP/IP
 Namespace no Caché

```

        System.out.println("Entre com dois argumentos:
Nome e Cpf");
    }
    catch (Exception e) {
        System.out.println(e.getMessage());
        e.printStackTrace();
    }
}
}

```

- Posicionar-se em **C:\TrabalhoJava\exemplo**;
- Compilar o aplicativo em java:
javac exemplo1a.java
- Executar o aplicativo em java (requer parâmetros):
java exemplo.exemplo1a Pedro 12345678900

Java – exemplo1b (acesso nativo)

O aplicativo do exemplo a seguir abre uma conexão com o Caché, também via Caché Java Binding e executa uma Query (da classe) que relaciona os objetos existentes na classe Pessoa:

```

package exemplo;
import COM.intersys.objects.*;
import COM.intersys.util.*;
public class exemplo1b
{
    public static final String CACHE_URL =
        "cn_iptcp:127.0.0.1[1972]:JAVA";
    static ObjectFactory factory = null;
    public static void main(String args[])
    {
        try
        {
            ResultSet rs;
            factory = new ObjectFactory(CACHE_URL);
            rs = new ResultSet(factory, "Pessoa", "Tudo");
            rs.execute();
            while(rs.next())
            {
                System.out.println("Nome:
                "+rs.getString(2));
            }
        }
    }
}

```

Cria um ResultSet com base na query "Tudo" definida na classe "Pessoa".

```
        System.out.println("CPF:
            "+rs.getString(3));
        System.out.print("\n");
    }
    rs.close();
}
catch (CacheException e) {
    System.out.println("Erro no cache: " +
        e.getMessage());
    e.printStackTrace();
}
catch (Exception e) {
    System.out.println(e.getMessage());
    e.printStackTrace();
}
}
}
```

- Posicionar-se em **C:\TrabalhoJava\exemplo;**
- Compilar o aplicativo em java:
javac exemplo1b.java
- Executar o aplicativo em java:
java exemplo.exemplo1b

Java – Exemplos de acesso via JDBC

- Aplicativo java **exemplo2a** insere um novo objeto na classe **Pessoa** via SQL;
- Aplicativo java **exemplo2b** executa uma pesquisa (query) na classe **Pessoa** via SQL.

Java – exemplo2a (acesso JDBC)

Similar ao exemplo1a, este exemplo a seguir também recebe dois valores de argumento (**Nome** e **CPF**) e cria um novo registro na classe **Pessoa**. Mas neste caso, o acesso é feito via JDBC usando SQL:

```
package exemplo;
import java.sql.*;
import java.io.*;

public class exemplo2a
{
    public static final String url =
        "jdbc:Cache://127.0.0.1:1972/JAVA";
    public static String sql;
    static Connection con = null;

    static {
        try {

            Class.forName("com.intersys.jdbc.CacheDriver");
        }
        catch (ClassNotFoundException e) {
            System.out.println("Driver do Cache nao
                encontrado.");
            System.out.println(e.getMessage());
            e.printStackTrace();
        }
    }

    public static void main(String[] args)
    {
        try {
            sql = "INSERT INTO Pessoa Set Nome=?,CPF=?";
            con=DriverManager.getConnection(url, "_SYSTEM", "sys");
```

Conexão via JDBC também requer endereço do servidor, porta e namespace.

Identifica o driver.

```
        PreparedStatement pstmt =
        con.prepareStatement(sql);
        pstmt.setString(1, args[0]);
        pstmt.setString(2, args[1]);
        pstmt.execute();
    }
    catch(SQLException e) {
        System.out.println(e.getMessage());
        e.printStackTrace();
    }
    catch (ArrayIndexOutOfBoundsException e)
    {
        System.out.println("Entre com dois argumentos:
        Nome e Cpf");
    }
    catch(Exception e) {
        System.out.println(e.getMessage());
        e.printStackTrace();
    }
    finally {
        try {
            con.close();
        }
        catch(Exception ignored) {}
    }
}
}
```

Logon com
usuário e senha.

- Posicionar-se em **C:\TrabalhoJava\exemplo;**
- Compilar o aplicativo em java:
javac exemplo2a.java
- Executar o aplicativo em java (requer parâmetros):
java exemplo.exemplo2a Carlos 44433322211

Java – exemplo2b (acesso JDBC)

Este exemplo executa um consulta SQL (Select) a classe **Pessoa** acessado o Caché via JDBC:

```
package exemplo;
import java.sql.*;
import java.io.*;
public class exemplo2b
{
    public static final String url =
        "jdbc:Cache://127.0.0.1:1972/JAVA";
    public static String sql;
    static Connection con = null;
    static {
        try {

Class.forName("com.intersys.jdbc.CacheDriver");
        }
        catch (ClassNotFoundException e) {
            System.out.println("Driver do Cache nao
                               encontrado.");
            System.out.println(e.getMessage());
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        try {
            sql = "SELECT Nome,cpf FROM Pessoa";
            con =
DriverManager.getConnection(url,"_SYSTEM","sys");
            Statement stmt = con.createStatement();
            ResultSet rs    = stmt.executeQuery(sql);

            while (rs.next()) {
                System.out.println("Nome: " +
                                   rs.getString(1));
                System.out.println("CPF: " +
                                   rs.getString(2));
                System.out.print("\n");
            }
            stmt.close();
        }
    }
}
```

```

        catch (SQLException e) {
            System.out.println(e.getMessage());
            e.printStackTrace();
        }
        catch (Exception e) {
            System.out.println("Ocorreu um erro: " +
                               e.getMessage());
            e.printStackTrace();
        }
        finally {
            try {
                con.close();
            }
            catch (Exception ignored) {}
        }
    }
}

```

- Posicionar-se em **C:\TrabalhoJava\exemplo;**
- Compilar o aplicativo em java:
javac exemplo2b.java
- Executar o aplicativo em java:
java exemplo.exemplo2b

Caché & Java – Alterações no Caché 5

- Na nova versão do Caché 5.0, que será lançada oficialmente em breve, algumas modificações na conectividade Java estão previstas:
 - O diretório padrão dos arquivos java (**C:\CacheSys\Java**) será alterado para **C:\CacheSys\Dev\Java\Lib;**
 - Os dois arquivos **.jar** serão unificados em apenas um de nome **CacheDB.jar.**