

Artículo

[Pierre-Yves Duq...](#) · Mar 10, 2021 Lectura de 5 min

Cómo migrar de Java Business Host a PEX

Con el lanzamiento de [PEX](#) en InterSystems IRIS 2020.1 e InterSystems IRIS for Health 2020.1, los clientes tienen una mejor forma de incorporar Java en las producciones que con el Java Business Host. PEX proporciona un completo conjunto de APIs para construir la interoperabilidad de los componentes y está disponible tanto en Java como en .NET. Java Business Host ha sido discontinuado y se retirará en una versión futura.

Ventajas de PEX

- Permite que los desarrolladores creen cualquier componente de producción tanto en Java como en .NET
- Se pueden transferir estructuras de mensajes más complejas entre los componentes
- Configuración simplificada
- Workflow de desarrollo simplificado, sin necesidad de ObjectScript.

El resto de este artículo se centra en cómo migrar el código existente de Java Business Host a PEX.

Resumen

Las clases y las interfaces que utiliza PEX son diferentes a las de Java Business Host (JBH). Aquí ofrecemos un resumen de las diferencias, pero la [documentación](#) recoge toda la información necesaria.

- [Cómo cambiar un Business Service](#)
- [Cómo cambiar una Business Operation](#)
- [Configuración](#)
- [Mensajes](#)
- [Logging](#)

Cómo cambiar un Business Service desde Java Business Host a PEX

Para crear un PEX Business Service, debes implementar `com.intersystems.enslib.pex.BusinessService` en vez de `com.intersystems.gateway.bh.BusinessService`.

El patrón de diseño utilizado por PEX para el Business Service ha cambiado de uno en el que se espera que el servicio inicie un subproceso para producir mensajes, a otro en el que el servicio implementa una función que es llamada periódicamente para producir mensajes.

En JBH, tu código se parecería a esto:

```
@Override
public boolean OnInit(Production p) throws Exception {
    production = p;

    if (messageThread == null) {
        Messenger messenger = new Messenger();
        messageThread = new Thread(messenger);
        messageThread.start();
    }
}
```

```
    }  
  
    return true;  
}
```

En PEX, solo necesitas implementar tres funciones:

```
public void OnInit() throws Exception {  
    // Initialization  
    return;  
}  
  
public Object OnProcessInput(Object messageInput) throws Exception {  
    // Here is where you call SendMessage() or SendMessageAsync()  
  
    return null;  
}  
  
public void OnTearDown() throws Exception {  
    // Shut down  
    return;  
}
```

También deberás cambiar la forma en que se utiliza la configuración, se entregan los mensajes y se hace logging. Hablaremos de eso más adelante.

Cómo cambiar una Business Operation desde Java Business Host a PEX

Para crear una PEX Business Operation, debes implementar `com.intersystems.enslib.pex.BusinessOperation` en vez de `com.intersystems.gateway.bh.BusinessOperation`.

El patrón de diseño para Business Operations es estructuralmente el mismo entre JBH y PEX, pero han cambiado los parámetros a dos puntos de acceso principales.

Cambios en OnInit()

En PEX, `OnInit()` no requiere de ningún parámetro.

Cambios en OnMessage()

En PEX, `OnMessage()` recibe un `Object` genérico en vez del `String` usado en JBH. Esto permite al autor de la producción transmitir cualquier tipo de mensaje que desee.

En JBH, tu aplicación pudo haber tenido este aspecto:

```
public boolean OnMessage(String message) throws Exception {  
    // Business logic here  
    return true;  
}
```

En PEX, el parámetro es un `Java Object` genérico, que se debe lanzar de forma adecuada, lo que permite transmitir mensajes más complejos que únicamente cadenas. Este es un ejemplo de cómo extraer una solicitud

que es una secuencia de archivos:

```
public Object OnMessage(Object request) throws Exception {
    com.intersystems.jdbc.IRISObject streamContainer = (com.intersystems.jdbc.IRISObject)request;
    com.intersystems.jdbc.IRISObject str = (com.intersystems.jdbc.IRISObject)streamContainer.get("Stream");
    String originalFilename = (String)streamContainer.get("OriginalFilename");

    Long contentSize = (Long)str.get("Size");
    String content = (String)str.invoke("Read", contentSize);

    // Business logic here

    return null;
}
```

También deberás cambiar la forma en que se utiliza la configuración, se entregan los mensajes y se hace logging. Hablaremos de eso más adelante.

Configuración

Se ha simplificado la declaración de la configuración.

La configuración en JBH se declaraba mediante una cadena SETTINGS y se obtenía a través de un código que se parece a algo como esto:

```
String setting = production.GetSetting("Min");
if (!setting.isEmpty()) {
    min = Integer.parseInt(setting);
}
```

En PEX, la configuración son solo campos para miembros públicos. Estos se completan automáticamente cuando la clase crea una instancia.

```
public int Min = 0;
```

Cualquier campo de un miembro público está disponible para que se establezca en una producción, en la medida en que el campo del miembro sea de un tipo básico de Java (String, int, etc.).

Mensajes

El envío de mensajes es más potente. En JBH, los mensajes se envían como cadenas. En PEX, los mensajes se envían como objetos- IRISObject, para objetos definidos en ObjectScript, o una subclase de com.intersystems.enslib.pex.Message, para clases definidas en Java.

En JBH, tu código se parecería a esto:

```
production.SendRequest(value.toString());
```

En PEX, sería algo como esto:

```
MyExampleMessageClass req = new MyExampleMessageClass("message to send");
SendRequestAsync(Target, req);
```

Logging

Todas las funciones de logging son similares, solo que se nombran de manera diferente.

En PEX, se registraría un mensaje informativo a través de LOGINFO()

```
LOGINFO("Received message");
```

Object Gateway

JBH necesitaba su propio portal. Con PEX, puedes utilizar un único portal de Java para todas tus necesidades de Java. O puedes utilizar varios portales. Depende de ti. Aquí encontrarás una buena [introducción al portal de Java](#).

Conclusión y comentarios

Si aún no has probado PEX, ¿a qué está esperando? PEX permite resolver un conjunto mucho más amplio de problemas empresariales con menos código, además de que ahora también puedes hacer cualquier cosa en .NET.

Si tienes alguna pregunta o problema para migrar tu aplicación de JBH a PEX, puedes contactar conmigo o con el Centro de Soporte Internacional (WRC).

[#.NET](#) [#Interoperabilidad](#) [#Java](#) [#Mejores prácticas](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-migrar-de-java-business-host-pex>