

Calling a C function `P func(P p)`.

1 Manuel's c2hs

declaration	result
<code>{#pointer P#}</code>	<code>type P = Ptr ()</code>
<code>{#pointer P newtype #}</code>	<code>newtype P = P (Ptr P)</code>
<code>{#pointer P foreign #}</code>	<code>type P = ForeignPtr ()</code> <code>withP p = withForeignPtr p</code>
<code>{#pointer P foreign newtype#}</code>	<code>newtype P = P (ForeignPtr P)</code> <code>withP (P p) = withForeignPtr p</code>

2 Gtk2Hs' c2hs

declaration	result
<code>{#pointer P#}</code>	<code>type P = Ptr ()</code>
<code>{#pointer P newtype #}</code>	<code>newtype P = P (Ptr P)</code>
<code>{#pointer P foreign #}</code>	<code>type P = ForeignPtr ()</code>
<code>{#pointer P foreign newtype#}</code>	<code>newtype P = P (ForeignPtr P)</code>

3 “The Right Thing” c2hs

declaration	result
<code>{#pointer P#}</code>	<code>type P = Ptr ()</code>
<code>{#pointer P newtype #}</code> <code>{#pointer P newtype #}</code>	<code>newtype P = P (Ptr P)</code> <code>newtype P = P (Ptr P)</code>
<code>{#pointer P foreign #}</code>	<code>type P = ForeignPtr ()</code>
<code>{#pointer P foreign newtype#}</code>	<code>newtype P = P (ForeignPtr P)</code>
<code>{#pointer P foreign(f) #}</code>	<code>type P = ForeignPtr ()</code>
<code>{#pointer P foreign(f) newtype#}</code>	<code>newtype P = P (ForeignPtr P)</code>