

# NOKIA

诺基亚论坛

## WAP Push 业务入门

版本 1.02 2001 年 12 月 12 日

本文内容可未经通知而修改，且所说明的只是本文中规定的产品。本文旨在为诺基亚网络公司的客户使用，且只适用于本文须遵守的协议，未经诺基亚网络公司的事先书面许可，不得以任何形式或手段复制或传播本文的任何内容。本文为专业及受过适当培训的人员使用而备，使用时，客户承担一切责任。诺基亚网络公司欢迎客户提出意见，以助于本文得以继续发展和改进。

关于提到的硬件或软件产品的适用性、能力或性能方面，本文中所述的信息或说明不具约束力，但应在诺基亚网络公司与客户所签协议中有规定。诺基亚网络公司已尽其最大的努力来保证本文所述的说明准确且无实质性错误和疏忽。但是，如有必要，诺基亚网络公司可对本文中可能未涉及的问题加以解释。

对于本文中的任何错误，诺基亚网络公司的责任只限于对其进行更正。诺基亚网络公司**在任何情况下都不对本文中的错误或任何损坏、附带事件或后果（包括货币损失）负责**，尽管这些情况可能是因为使用本文件或其中信息而引起。

根据相关法律，本文及其中介绍的产品受版权保护。

诺基亚标志是诺基亚公司的注册商标。

本文中提到的其它产品名称可能是其相应公司的商标，提到其产品的目的只为进行识别。

版权©诺基亚公司 2001。版权所有。

目 录

1	读者与范围 .....	4
2	如何提交 WAP Push 业务提示(SI) .....	5
2.1	提交的内容是多段文件 .....	7
2.2	内容类型 .....	7
2.3	Puh-id .....	7
2.4	源引用属性 .....	7
2.5	业务提示参数 .....	8
2.6	回应 .....	8
2.7	PPG 通知 .....	8
3	库的使用 .....	10
	参考文献 .....	11
	附录 A. 发送业务提示的简单 servlet .....	13
	附录 B. 发送业务提示的 Stylesheet .....	17

# 1 读者与范围

本文是为致力开发移动业务的应用程序开发者编写的，并且这些移动业务采用了诺基亚Activ Alert和/或诺基亚Artuse 网关来发送WAP Push业务提示。

本文涉及的范围包括提供与向WAP Push代理网关提交Push请求有关的基本信息。本文不会替代任何其他的WAP Push资料或使这些资料被人遗忘。

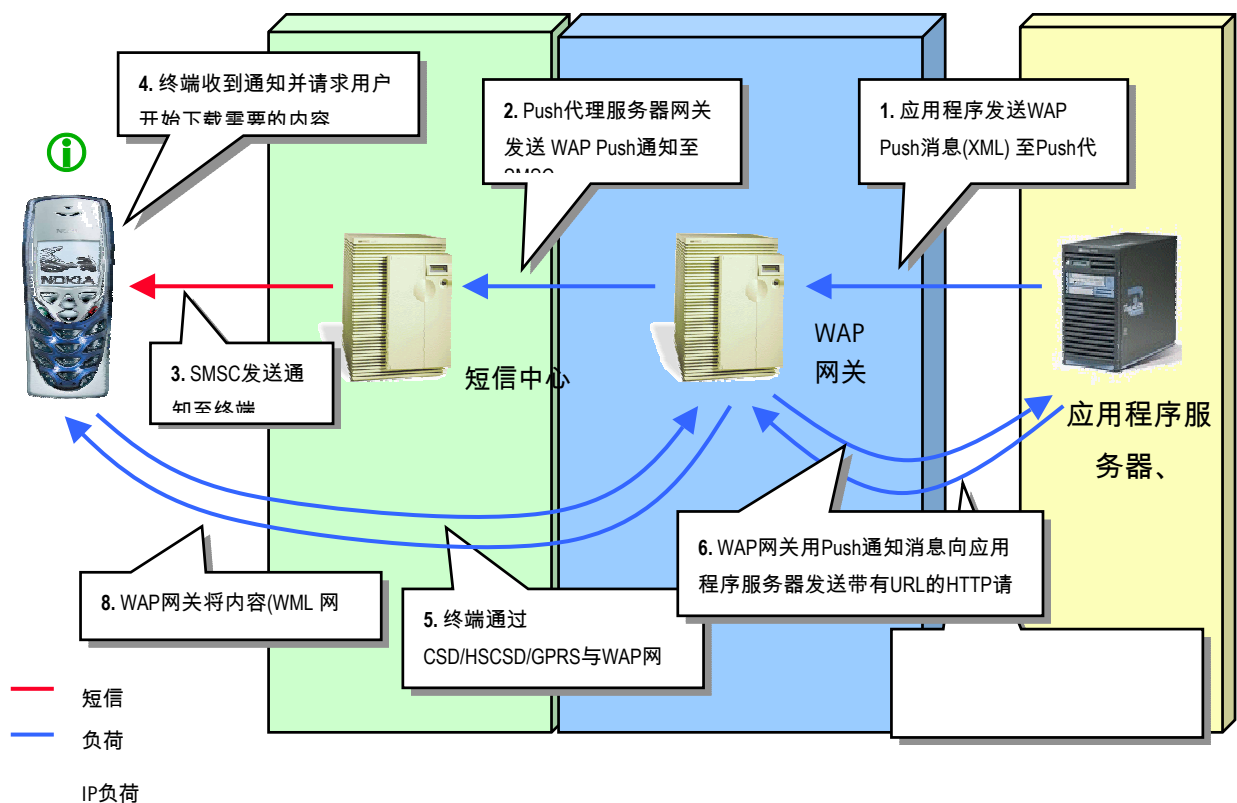
## 2 如何提交 WAP Push 业务提示(SI)

需要发送WAP Push业务提示 ( SI ) 的业务应用程序应该首先向WAP Push网关 ( PPG ) 发送 HTTP POST。应用程序开发者应该注意：

- a. 请求的正文格式：它必须遵从WAP规定的格式。
- b. 鉴权：WAP PPG可能强行加入一些接入控制功能，并且对发往用户的SI的实际发送进行控制。
- c. PPG对通知的可能处理。

下面将给出一个关于请求和通知的范例，并且将其中的关键部分加黑。

下图给出了Push消息是如何从应用程序发送到PPG和终端的过程。



下面是一个从应用程序到PPG之间提交业务提示的范例：

```
POST /PGW/ HTTP/1.1
Authorization: Basic xyzabc123
Content-type: multipart/related; boundary=multipart-boundary
User-Agent: Javal.3.1
Host: 195.255.249.57:5080
Accept: */*
Connection: keep-alive
Content-length: 971
```

```
--multipart-boundary
Content-type: application/xml
```

```
<?xml version="1.0"?>
<!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 1.0//EN"
"http://www.wapforum.org/DTD/pap_1.0.dtd">
<pap product-name="Nokia Push Initiator">
<push-message
  push-id="SomePushApp1013753552819"
  source-reference="SomePushApp"
  progress-notes-requested="false"
  ppg-notify-requested-to=
    "http://123.123.123.123/push/notificationInbox.jsp">
    <address address-value="WAPPUSH=+358454410926/TYPE=PLMN@null"/>
    <quality-of-service
      bearer-required="false"
      network-required="false"
      delivery-method="notspecified"
      priority="low"/>
    </push-message>
</pap>
```

```
--multipart-boundary
Content-Type: text/vnd.wap.si
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"
"http://www.wapforum.org/DTD/si.dtd">
<si>
  <indication href
    ="http://123.123.123.123/pushservice/actionmanager"
    si-id="SomePushApp1013753552819"
    action="signal-high"
    created="2002-02-15T08:12:32Z">
    Some Push Application
  </indication>
</si>
```

```
--multipart-boundary--
```

## **2.1 提交的内容是多段文件**

由于提交的 SI 是以多段文件形式呈现的，HTTP POST 的内容类型必须设置为相关的多段类型。其分界应该是不能在其它内容部分出现的线段。

提交内容的每部分是以两个破折号加一段定义的分界线开始的。因此这部分的内容类型头 ( Header ) 紧跟在这部分的分隔符 ( Separator ) 之后。提交内容的最后部分是以一个分隔符结束的，此分隔符为在分界线前后各有两个破折号。

## **2.2 内容类型**

必须注意提交内容的开始部分是“application/xml”类型，但实际的业务提示却是“text/vnd.wap.si”内容类型。

## **2.3 Puh-id**

应用程序能够产生一个 push-id，此 push-id 可以用来唯一地识别提交内容。Push-id 可以为任意字符串。应用程序产生的 push-id 可以基于应用程序的名称、日期和时间以及连续数字等。此相同的数字也可以用作 SI 的 si-id。

## **2.4 源引用属性**

源引用属性可以用作识别应用程序过滤功能的方法。运营商和最终用户能够禁止某些业务中的 push 消息。因此，必须与 PPG 的运营商协商确定此属性的内容。

注意有些 PPG 要求应用程序在发送 HTTP 请求时，授予自己一个授权头 ( Authorization header )。此授权头包含了 PPG 运营商认可的应用程序或帐号的名称和密码。

例如，可以用 Java 将授权头设置如下：

```
URLConnection connection = _pgw.openConnection();  
connection.setRequestProperty("Authorization", "Basic xyzabc123");
```

## 2.5 业务提示参数

业务提示请求的实际参数如下列蓝色部分所示。这些通常是从一个请求变成另一个请求时的参数。

提示接收者的 MSISDN 号码**+358454410926** 是在<pap>请求之中。实际的 SI 包含一个具有 URL `<a href>` 属性的 `<indication>` 单元，例如 <http://123.123.123.123/pushservice/actionmanager>。其它属性也不能忽视。终端用户能够看到<indication element>的文本文字，这些文字描述了什么内容将被 SI 下载，例如：**Some Push Application。**

## 2.6 回应

从 PPG 至提交的回应如下所示：

```
HTTP/1.1 202 Unknown reason
Server: Netscape-Enterprise/4.1
Date: Fri, 15 Feb 2002 06:12:07 GMT
Content-type: application/xml
Transfer-Encoding: chunked

<?xml version="1.0"?>
<!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 1.0//EN"
"http://www.wapforum.org/DTD/pap_1.0.dtd">
<pap product-name="Nokia Activ Alert Push Subsystem">
  <push-response
    push-id="SomePushApp1013753552819"
    reply-time="2002-02-15T06:12:08UTC">
    <response-result
      code="1001"
      desc="Accepted for processing">
    </response-result>
  </push-response>
</pap>
```

## 2.7 PPG 通知

提交 Push 请求的应用程序可以要求 PPG 来通报 Push 消息的生命状态。这是通过设置一个 URL 的 `ppg-notify-requested-to` 属性来实现的，此 URL 指向一个能够处理通知的脚本语言程序 (script)。

实际的通知如下所示：



```
<?xml version="1.0"?>
<!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 1.0//EN"
    "http://www.wapforum.org/DTD/pap_1.0.dtd">
<pap>
  <resultnotification-message
    push-id="Cowbuy200109101001"
    message-state="delivered"
    code="1000"
    desc="Push Successful"
    received-time="2001-09-10T12:24:45UTC"
    event-time="2001-09-10T12:24:45UTC">
    <address address-value="WAPPUSH=0454410916/TYPE=PLMN@null"/>
    <quality-of-service delivery-method="unconfirmed"/>
  </resultnotification-message>
</pap>
```

此通知说明了即使 SI 发送到短消息中心，也不意味着 SI 一定能够到达终端上。

## 3 库的使用

使用库是非常方便的，此库包含了 push 请求。诺基亚为一些初始的 push 应用程序开发者提供了一个 Java 软件包。诺基亚 Activ Alert 包含此库，并且它可从 <http://www.forum.nokia.com> 网址获得。Activ Alert 的用户指南中有一章介绍如何对库进行编程。

一个更为灵活的方法是利用 XSLT 转换。此方法是将提交内容的关键信息放入一个 XML 文件，然后，由 XSL style sheet 将此文件转换成相应的 PAP 请求。之后，此请求被送到 Push 网关。参见附录 A 中一个能够完成上述操作的简单 servlet 范例，以及附录 B 中与此 servlet 配合工作的 style sheet。

## 参考文献

1. 多段内容的 RFC ( RFC for multipart content ) 。
2. WAP PAP 技术规范 ( WAP PAP spec ) 。
3. WAP SI 技术规范 ( WAP SI spec ) 。
4. Activ Alert 用户指南 ( Activ Alert User Guide ) 。



## 附录 A. 发送业务提示的简单 servlet

```
package com.nokia.pgw;

import javax.servlet.ServletException;
import java.io.*;
import java.net.*;
import java.text.SimpleDateFormat;
import javax.xml.transform.Transformer;
import javax.servlet.http.*;
import org.dom4j.*;
import org.dom4j.io.*;

/**
 * The PushSIServlet submits a push request
 * with a Service Indication to a Push Gateway
 *
 * Servlet requires three parameters provided in HTTP request:
 * ms = MSISDN to send push message
 * href = URL of push message service
 * msg = Message visible to user of push message
 *
 * @Nokia, 2001
 */

public class PushSIServlet extends HttpServlet {

    // MSISDN parameter name from request
    private static String MSISDN_PARAM = "ms";
    // URL parameter name from request
    private static String HREF_PARAM = "href";
    // Message parameter name from request
    private static String MESSAGE = "msg";
```

```
// Application name
private String _app_name = "MyPushApp";
// Notification URL initial set-up, can be set up in servlet initialisation parameters
private String _notification_URL = "http://localhost/push/notificationInbox.jsp";
// XSLT Transformer object
private Transformer _siTransformer;
// Push Gateway URL address
private URL _pgw;

public void init()
    throws ServletException
{
    // Load the SI stylesheet as Transformer
    try {
        javax.servlet.ServletContext context = getServletContext();
        _siTransformer = loadStylesheet(context.getRealPath("si.xsl"));
        _pgw = new URL(getInitParameter("pgw"));
        _app_name = getInitParameter("app-name");
        _notification_URL = getInitParameter("notification-URL");
    }
    catch (MalformedURLException e) {
        throw new ServletException("PGW URL is malformed", e);
    }
}

public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    doPost(request, response);
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    URLConnection connection = _pgw.openConnection();
```

```
connection.setRequestProperty("Authorization", "Basic xyzabc123");
connection.setRequestProperty("Content-type", "multipart/related; boundary=multipart-boundary");
Document submission = DocumentHelper.createDocument();
Element si = submission.addElement("si"); // We create a XML doc with a si root element
si.addAttribute("push-id", _app_name + System.currentTimeMillis());
si.addAttribute("URL", getParameter(HREF_PARAM, request));
si.setText(getParameter(MESSAGE, request)); // The message that the user will see
si.addElement("addressee").addAttribute("address", getParameter(MSISDN_PARAM, request));
_siTransformer.setParameter("application-name", _app_name);
_siTransformer.setParameter("notification-URL", _notification_URL);
SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd'T'hh:mm:ss'Z'");
String created = formatter.format(new java.util.Date());
_siTransformer.setParameter("time", created);
connection.setDoOutput(true);
submit(submission, _siTransformer, connection.getOutputStream());
PrintWriter writer = response.getWriter();
BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
String inputLine;
while ((inputLine = in.readLine()) != null)
    writer.println(inputLine);
}

private void submit(Document xmlDoc, Transformer stylesheet, OutputStream out)
    throws ServletException
{
    try {
        javax.xml.transform.stream.StreamResult result = new javax.xml.transform.stream.StreamResult(out);
        stylesheet.transform(new DocumentSource(xmlDoc), result);
    }
    catch (javax.xml.transform.TransformerException e) {
        throw new ServletException("XSLT problem", e);
    }
}

private void printXML(Node xmlNode, Transformer stylesheet, javax.servlet.http.HttpServletResponse
```

```
        response) throws ServletException
    {
        try {
            XMLWriter xwriter = new XMLWriter(response.getOutputStream());
            xwriter.write(xmlNode);
        }
        catch (IOException e) {
            throw new ServletException("XSLT Response problem", e);
        }
    }

private Transformer loadStylesheet(String fullPathName)
    throws ServletException
{
    try {
        Transformer transformer = javax.xml.transform.TransformerFactory.newInstance().newTransformer
            (new javax.xml.transform.stream.StreamSource(fullPathName));
        return transformer;
    }
    catch (javax.xml.transform.TransformerConfigurationException e) {
        throw new ServletException("XSLT Transformer problem", e);
    }
}

protected String getParameter(String name, HttpServletRequest request)
{
    String value = request.getParameter(name);
    if (value == null) value = "";
    return value;
}
}
```



## 附录 B. 发送业务提示的 Stylesheet

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <xsl:output
    method="xml"
    indent="yes"
    encoding = "UTF-8"
    omit-xml-declaration="yes"/>

  <xsl:param name="application-name" />
  <xsl:param name="notification-URL" />
  <xsl:param name="time" />

  <xsl:template match="*[@push-id]">
--multipart-boundary
Content-type: application/xml

<xsl:text disable-output-escaping="yes">&lt;?xml version=&quot;1.0&quot;?&gt;</xsl:text>
<xsl:text disable-output-escaping="yes">&lt;!DOCTYPE pap PUBLIC &quot;-//WAPFORUM//DTD PAP 1.0//EN&quot;
&quot;http://www.wapforum.org/DTD/pap_1.0.dtd&quot;&gt;</xsl:text>

  <pap product-name="Nokia Push Initiator">
    <xsl:element name="push-message">
      <xsl:attribute name="push-id"><xsl:value-of select="@push-id"/></xsl:attribute>
      <xsl:attribute name="source-reference"><xsl:value-of select="$application-name" /></xsl:attribute>
      <xsl:attribute name="progress-notes-requested">false</xsl:attribute>
      <xsl:if test="$notification-URL">
        <xsl:attribute name="ppg-notify-requested-to">
          <xsl:value-of select="$notification-URL" />
        </xsl:attribute>
      </xsl:if>
    </xsl:element>
  </pap>
</xsl:template>
</xsl:stylesheet>
```

```
    </xsl:if>
    <xsl:apply-templates select="addressee" />
    <quality-of-service
      priority="low"
      delivery-method="notspecified"
      network-required="false"
      bearer-required="false"/>
  </xsl:element>
</pap>
```

--multipart-boundary

Content-Type: text/vnd.wap.si

```
<xsl:text disable-output-escaping="yes">&lt;?xml version="1.0" encoding="ISO-8859-1" ?></xsl:text>
<xsl:text disable-output-escaping="yes">&lt;!DOCTYPE si PUBLIC "http://www.wapforum.org/DTD/si.dtd" //WAPFORUM//DTD SI 1.0//EN" "http://www.wapforum.org/DTD/si.dtd" //WAPFORUM//DTD SI 1.0//EN" ></xsl:text>
```

```
<si>
  <xsl:element name="indication">
    <xsl:attribute name="href"><xsl:value-of select="@URL" /></xsl:attribute>
    <xsl:attribute name="si-id"><xsl:value-of select="@push-id" /></xsl:attribute>
    <xsl:attribute name="action">signal-medium</xsl:attribute>
    <xsl:attribute name="created"><xsl:value-of select="$time" /></xsl:attribute>
    <xsl:value-of select="." />
  </xsl:element>
</si>
```

--multipart-boundary--

```
</xsl:template>
```

```
<xsl:template match="addressee">
  <xsl:element name="address">
    <xsl:attribute name="address-value">WAPPUSH=
```

```
        <xsl:value-of select="@address" />
      /TYPE=PLMN@null</xsl:attribute>
    </xsl:element>
  </xsl:template>

</xsl:stylesheet>
```