



Code less.
Create more.
Deploy everywhere.

Qt 集成 WebKit

渲染、使用和扩展网络技术

概要

开源 WebKit 浏览器引擎与 Qt 4.5 应用程序和 UI 框架的集成模糊了本地应用程序与网络内容之间的界限。有了此类混合模式的服务，就可以高效地构建和发布很难或无法仅通过网络 API 或本地开发环境单独开发的功能。由于融合了网络技术的简易性和 Qt 应用程序框架的强大功能，设计人员和编码人员可专注于自己最擅长的部分。本白皮书列举了一个简单的设计示例：可访问 Twitter 服务的跨平台应用程序。该示例通过三个阶段，详细说明了如何使用集成了 WebKit 的 Qt 4.5，开发出渲染、集成和扩展网络服务的本地应用程序。

目录

1. 奠定基础	3
2. 渲染所有组成部分	4
2.1 获得网络应用程序的性能优势	6
3. 集成应用程序和网络服务	7
3.1 自动验证	7
3.2 利用网络服务	7
4. 扩展性能	8
4.1 编写通过 OBJECT 标签的访问插件	9
4.2 通过 JavaScript 使用本地功能	9
4.3 融为一体	10
5. 双向扩展	11
6. 培养创意思维	12
7. 开始入门	12

1. 奠定基础

开源 WebKit 浏览器引擎与 [Qt 4.5 应用程序框架](#) 的集成模糊了应用程序与网络内容的差别。它可以访问所有网络服务、访问所有可在本地编程环境获得的性能和控件，并可通过最大化网络服务范围的跨平台框架进行构建。有了此类混合模式的服务，就可以高效地构建和发布很难或无法仅通过网络 API 或本地开发环境开发的功能。在开始讨论集成的优势之前，很有必要了解要集成的所有组成部分。

Qt 是跨平台的应用程序框架。Qt 1.0 于 1994 年发布，在此之后 Qt 逐渐发展成为开发高性能应用程序的标准框架，这些应用程序只需使用单一的源代码就可跨 Microsoft Windows、Mac OS X、Linux、所有主要 Unix 系统、以及嵌入式 Linux 和 Windows CE 移植。已有来自不同行业的 5,000 多家公司在使用 Qt，使用它的开源开发人员大约有 250,000 人。一些最知名的应用程序是使用 Qt 框架构建，例如 Adobe 使用 Qt 构建其 Photoshop Elements 软件，Google 使用 Qt 构建 Google Earth，Skype 使用 Qt 构建其 Linux 客户端。

WebKit 是开源网络浏览器引擎。它完全符合标准，支持与当前网络内容兼容所需的所有功能。它完全支持 HTML 4.01、XHTML 1.1、XML、CSS 2.1、JavaScript 1.5、大部分 SVG 1.2、部分 CSS 3、HTML editing、HTML canvas、AJAX、XSLT 和 XPath 1.0。

WebKit 是以 LGPL (和部分 BSD) 方式授权的，此类授权适合将 WebKit 集成到商业框架中。WebKit 开发是采用开源形式的，对诺基亚和 Qt Development Frameworks 项目的未来发展方向有很大影响，也促使我们为 webkit 开发所需的功能不断努力。

WebKit 项目是 KDE's KHTML 浏览器引擎的一个分支。诺基亚和 Qt Development Frameworks 雇用了大量从事过 KHTML 工作的关键开发人员，其中包括 Qt Development Frameworks 的研发主管 Lars Knoll，他从 1998 年至 2003 年期间一直主管 KHTML 项目并对其进行维护。因此，我们对其源代码有非常深入的了解。

许多主要公司已采用 WebKit 作为其浏览器解决方案的首选基础。WebKit 是 Apple 的 Safari 浏览器、Google Chrome 和上百万手机用户使用的 S60 浏览器的引擎。

Qt 应用程序框架和 WebKit 浏览器引擎的集成为开发团队提供了两者的精华所在。因此，设计人员可使用熟悉的网络流程，而编码人员则可专心编写应用程序功能。这样的集成通过在本地应用程序中渲染网络内容，提供了本地和网络混合的用户体验。这样，本地应用程序就可通过编程与网络服务进行互动，例如自动认证过程。开发人员还可通过网络服务扩展本地应用程序的功能，以及使用一些本地环境中特有的功能来扩展网络服务。

- 有关更多信息，可通过 <http://bit.ly/RJlrf> 在线获得
- 有关 Qt 和 WebKit 操作的视频，请查看 <http://bit.ly/Bt6Co>
- 请从以下网址下载 Qt 4.5 (LGPL 授权和商业评估版) 的完整源代码 <http://bit.ly/17Ca8j>

2. 渲染所有组成部分

为了详细说明使用此框架可实现哪些功能，我们会逐步开发一个集成本地功能和网络技术的示例应用程序。我们先从基本功能开始：在本地应用程序内部渲染网页。我们会逐步添加功能，其中包括用户自动登录服务、从服务器提取信息以及通过非默认服务提供商提供的方式为用户提供信息。最后，我们会演示如何合并多种服务和本地功能，例如从网络服务提取地址并使用其他服务在地图上标记这些位置。最终的结果可能未必与您要开发的下一个项目相符，但却足以详细说明 QtWebKit 可支持的多种可能。

我们首先从渲染 Twitter 服务的主页开始。以下是 Twitter 主页的主要部分：

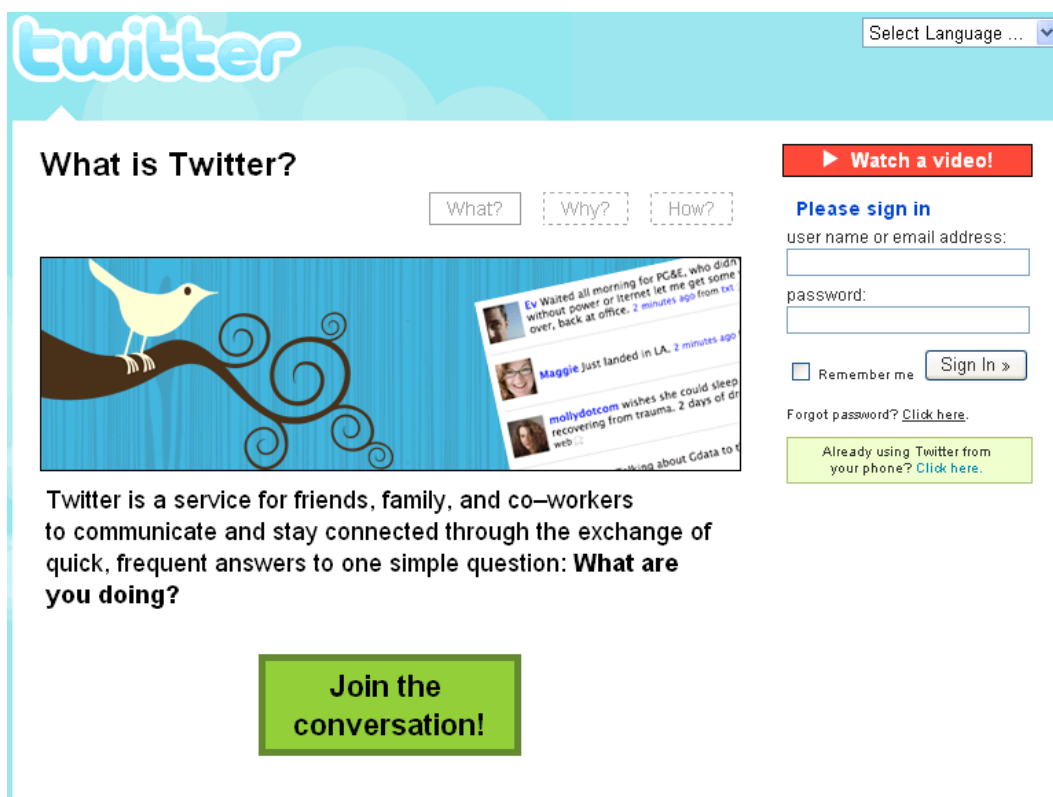


图 1：Qt 4.5 的 WebKit 引擎可处理除基础 HTML 外的更多内容。它可以处理 JavaScript、CSS 和所有其他由成熟的网络服务所需的技术。

要在用 Qt 4.5 开发的应用程序中渲染（显示）此网页，只需三行代码：

```
QWebView w;
w.load(QUrl("http://twitter.com/"));
w.show();
```

第一行实例化用于显示网页并与其互动的 `QWebView` 类。该类提供了常用的功能，如加载特定的 URL、设置、历史记录和网页对象。它还提供包括后退、向前和重新加载在内的基本浏览功能。`QWebView` 是 GUI 组件，因此可在 GUI 中单独使用。

后两行代码指示 `QWebView` 加载 URL 并显示，这样就得到一个可与网站互动的基本窗口。

完全再现网络互动需要不仅仅是显示静态文本和图形。例如，Twitter 网页包括使用 WebKit 引擎处理的内嵌 CSS 语句：

```
<style type="text/css">

    body { background: #9ae4e8
url(http://assets2.twitter.com/images/bg.gif) fixed no-repeat top
left; }

    ...

    #flash { background-image:
url('http://static.twitter.com/images/girl.gif'); }

</style>
```

WebKit 还可处理 JavaScript，例如处理用户在“加入对话”的对话框上单击操作的此类简短脚本。

```
<script>utmx_section("Signup Button")</script>
```

与 Qt 4.5 集成的 WebKit 引擎支持与当前网络内容兼容所需的所有功能。它完全支持 HTML 4.01、XHTML 1.1、XML、CSS 2.1、JavaScript 1.5、大多数 SVG 1.2 特性、部分 CSS 3 特性、HTML editing、HTML canvas、AJAX、XSLT 和 XPath 1.0。

并非所有浏览器引擎都具有符合嵌入 WebKit 标准的兼容性。可通过简单的方式对浏览器引擎进行全面测试。只需将浏览器链接至 <http://acid3.acidtests.org/>，观看动画，然后查看基于基准图像的最终结果。动画应流畅，并且测试数值应达到 100/100。在所见画面之后，有大量的 JavaScript 代码在对浏览器进行 100 项与你所见的 100 个动画步骤相关的子测试。整套测试检验了 ECMAScript、DOM Level 2、可扩展矢量图形 (Scalable Vector Graphics, SVG)、XML 和 data: URIs。它测试了 HTML 表格和表单在由脚本和 DOM2 HTML 控制时的表现。

还有更直接的方法查看 Qt 4.5 与 WebKit 集成在渲染您的网络内容时的效果。[下载免费的 Qt SDK](#)，创建一个简单的网络应用程序，然后将集成的浏览器链接至您要渲染的 URL

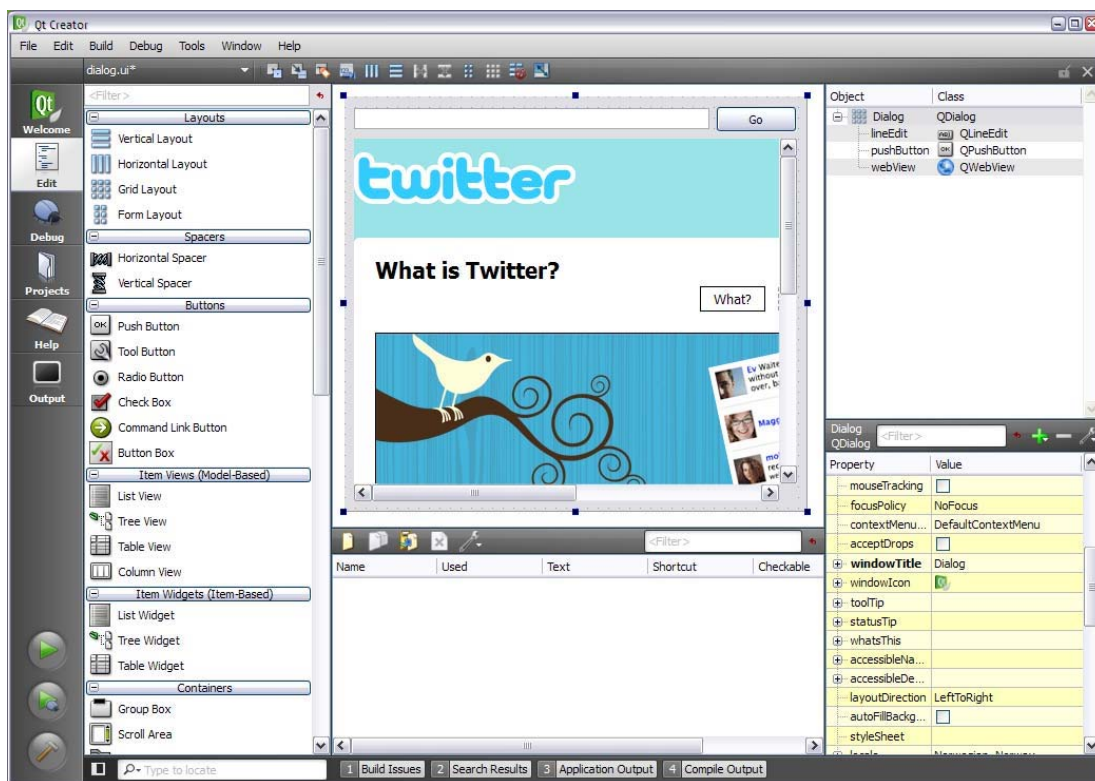


图 2：Qt SDK 包括了 Qt Creator，它是可用来在开发环境中直接预览实时网络集成情况的 IDE。

2.1 获得网络应用程序的性能优势

我们使用浏览器时，通常情况下，渲染的内容放置在某处的服务器上。如果使用 Qt 和 WebKit 浏览器引擎，我们就会有更多的灵活性。您可以增加本地缓存来提高响应时间，甚至还可以在您的应用程序中动态创建网络视图。

要提高响应时间，您可以增加缓存系统。QDesktopServices::storageLocation 中全新的 CacheLocation 枚举函数可处理在跨平台情况下，指定存储位置的复杂情况。包含的 QNetworkDiskCache 非常简单，并且完全可满足大多数应用程序的需求。但是，如果作业调用了用于小型设备的高度优化缓存，或者您需要利用其它缓存系统，则可使用 QabstractNetworkCache 接口，这样您就可以自己实现并使用定制的缓存系统。

您只需以下四行代码就可增加本地缓存来提高响应时间：

```
QNetworkDiskCache *diskCache = new QNetworkDiskCache(this);
QString location =
QDesktopServices::storageLocation(QDesktopServices::CacheLocation
);
diskCache->setCacheDirectory(location);
networkAccessManager->setCache(diskCache);
```

您甚至可以让程序自动生成用于报表、存档和帮助功能的内容。基本想法就是对 Twitter 示例简单扩展：

```
QWebView w;

w.setHtml("<h1>Hello World!</h1>");

w.show( );
```

3. 集成应用程序和网络服务

部分网站提供的不仅有内容，还有多种服务。网络服务的内容从货币汇率信息到企业应用程序接入外部网络的连接，多种多样。借助 Qt WebKit 集成，您可在应用程序中使用这些网络服务。让我们返回 Twitter 示例，实现自动化用户验证，然后考虑一下专为开发人员设计的网络服务。

3.1 自动化验证

以下三行代码就可创建 Twitter 的认证请求：

```
QNetworkRequest request("http://twitter.com");

req.setRawHeader("Authorization", basicBase64);

QNetworkReply *reply = manager->get(request);
```

Qt 中的网络 API 支持异步网络操作，包括网络服务的认证请求。以上代码创建了头字段 **Authorization**，包含了采用 **base64** 位编码的用户名和密码。然后，使用实例化的 **Net Manager**，通过调用 **get()** 提交请求，接着请求被发送至网站。

其他网站要求提交表单或上载数据。使用 QtWebKit 可采用与上述登录情景类似的方式，您可设置不同的头信息值，然后通过 **post()** 或 **put()** 提交至网站。

以下方法采用了从服务器接收回复的句柄。该句柄会检查结果数据，验证是否已成功登陆。此方法还适合与网络服务进行进一步互动，例如获取 XML 形式好友列表。

```
void requestFinished(QNetworkReply *r)

{

    if(authenticationSuccessful(r))

        webView->load(homeURL);

}
```

3.2 利用网络服务

此方法可扩展至通过 URL 或 JavaScript API 方式获得的网络服务，甚至是普通用户和标准浏览器无法访问的特定服务。以 **Google Maps** 为例，它提供了输入街道地址可得到该地经度和纬度坐标的服务。这种处理方法称为地理编码，是在

地图上确定个人位置的初步尝试。(大多数核心地图服务要求输入坐标，而不是地址。)

该代码使用了 Google 服务，该服务可接受地址、并返回 XML 格式的坐标以及要求输入可使用 API 的密钥。我们稍后会使用它为我们 Twitter 上的朋友在地图上一定位。

```
void geoCode( )
{
    QString requestStr =
        "http://maps.google.com/maps/geo?q="
        +address+ "&output=xml&key=GOOGLE_MAPS_KEY"
    QNetworkRequest req(requestStr);
    manager->get(req);
}
```

在访问通过 URL 或 JavaScript API 实现的网络服务时，都可以采用此方法。部分示例包括：

- 可通过 <http://www.salesforce.com/> 获得的商业服务，包括 Sales Force Automation (销售人员自动化) 和 Customer Relationship Management (客户关系管理)。
- <http://www.timeanddate.com/worldclock/> 的 **Time Zone translation** (时区转换)
- 包括 <http://www.ups.com> 在内的大多数运输公司采用的 **Package Tracking** (包裹跟踪)

4. 扩展性能

HTML、CSS 和 JavaScript 这三款网络利器简便易用，但各有不足之处。借助 Qt WebKit 集成，您可以通过构建定制浏览器扩展您应用程序的功能，实现那些普通的 HTML 开发人员所未想过的功能。

我们再次返回 Twitter 示例，这次设计可实现以下功能的跨平台网页：

1. 用户自动登录至已知的 Twitter 帐户。
2. 将用户的好友列表加载至互动控件。

设计标准 (1) 很简单，就是使用标准的 HTTP。我们在前面的章节已开发了该示例。

设计标准 (2) 比较有挑战性，是使用网络技术。Qt WebKit 集成可解决此难题，方法是可将本地功能暴露在网络环境中。

我们先看一下结果，然后再讨论过程。以下是一个网络视图，包含了嵌入用户好友列表的 UI Widget，以及自动显示选定 Tweet 的内容和作者的显示字段：



图 3：C++ 开发人员构建了可浏览用户好友列表的 UI Widget，然后网络开发人员将该 Widget 使用 <OBJECT> 标签 添加至其网页中。

很显然，此处的 UI Widget 是不会被 HTML 标准委员会承认的。不过，使用 Qt 和 WebKit，您可以在每个应用程序的基础上，通过编写自己的浏览器插件，扩展网络环境的功能。利用插件可将在本地环境可使用的功能用于网络应用程序

有两种方式可在网络应用程序中包括本地功能：编写通过 OBJECT 标签访问的本地对象插件，或者通过 JavaScript 调用本地功能。

4.1 编写通过 OBJECT 标签的访问插件

通过 C++ 方法 QWebPage::createPlugin 使用本地功能。您可完全控制为您的环境进行的 classid 字符分配。

```
QObject* MyPage::createPlugin(const QString & classid, ...)
{
    QWidget *widget=0;

    if (classid=="MyContacts")
        widget = new MyContacts;

    return widget;
}
```

接着，您就可通过 OBJECT 标签访问本地功能。

```
<OBJECT type="application/x-qt-plugin" classid="MyContacts"/>
```

4.2 通过 JavaScript 调用本地功能

使用 QWebFrame::addToJavaScriptWindowObject 可将本地对象添加至 JavaScript 环境。这意味着 Qt 应用程序框架提供的任何方法或您使用 Qt 应用程序框架构建的任何本地对象都可通过 JavaScript 访问。

4.3 融为一体

只有当您将所有这些成分融合在一起时，混合模式设计才会具有强大功能。通过扩展 JavaScript 代码的能力来处理程序，在本例中，使其具有以下能力：登录 Twitter 服务、提取好友列表以及在列表中增加好友。通过编写浏览器插件并借助 OBJECT 标签访问该插件来提供定制 UI 组件。

以下是代码概述：

```
<script>

function init( ) {

    if(twitter.login( )) // call C++ function

        loadFriendList( ); // XMLHttpRequest

}

function handleReply( ) {

    // ... handle reply ...

    twitter.addFriends(request.responseText);

}

</script>



<object classid="MyContacts">

<object classid="MyContactView">
```

请注意我们是如何使用 Twitter 类进行登录的。您可以将任何 Qt 本地对象用于网络环境，并且其所有属性和可见方法都可通过 JavaScript 调用。

好友列表是通过 AJAX-API XMLHttpRequest 加载的。网络服务器返回相关数据后，我们可将其传递至已应用于网络环境的本地对象。该本地对象就会装入好友列表。与好友列表互动的操作会提取相关数据，并可用来显示选定 Tweet 的信息。

使用这些工具可通过 JavaScript 控制初始化顺序和基础应用程序架构。在本示例中，我们需要登录至 Twitter 服务，然后加载好友列表。

该示例显示了在网络环境中融入本地控件的简便之处，网络开发人员或图形设计人员就可对其进行创建、设计和编排。您甚至可以将在本示例中使用的 HTML 放置在服务器端，这样在无需更新二进制执行文件的情况下即可动态更改本地应用程序的外观。

5. 双向扩展

到目前为止，都是讨论在网络环境中实现本地功能。实际上就是在 HTML 和 JavaScript 内部执行 C++ 功能。最后一个疑问就是在 C++ 应用程序中执行 JavaScript 功能的机制。这样，我们就可开发融合多种网络服务和本地功能优势的应用程序。

让我们再次回到 Twitter 示例，来完成将目前已开发的所有元素都融合在一起的任务：

1. 用户自动登录至已知的 Twitter 帐户。
2. 将用户的好友列表加载至互动控件。
3. 使用 Google Maps 在地图上标出选定好友的位置。

以下是最终结果：



图 4：该网络视图融合所有已开发的元素。JavaScript 的功能已进行扩展，用于自动化 Twitter 认证和访问 Google Maps 地理编码服务。浏览器插件创建了好友列表 widget。最后，二进制执行文件可控制 JavaScript 按期望顺序执行。

该地图显示了在用户选定 John Gruber 的 Tweet 时 John 所在的位置。一旦选定 Tweet，本地代码就会从记录中提取地址、接着访问地理编码服务进行坐标转换，然后将坐标传递给地图服务以便生成地图。

如果一个或多个网络服务仅可通过 Java API 访问，那么本地代码可使用 `QWebFrame::evaluateJavaScript()` 对 JavaScript 进行调用。

6. 培养创意思维

集成 Qt 应用程序框架和 WebKit 浏览器引擎的真正强大之处在于其带来的无限创意。有强大的本地代码做后盾，并利用跨平台框架平台，从而最大化你的网络服务范围，可将网络服务任意进行组合。如果您还不清楚这对您的业务的重要性，以下几点或许对您有所启发。

- 利用您的销售数据、地理编码服务和地图服务可以绘制销售地理区域。
- 利用有关卡车库存的数据和线路安排服务使服务卡车成为仓库的一部分，这样货运人员无需返回基地便可互相交接。
- 通过数据库技术，连接 Salesforce.com、企业人员列表以及私人联系列表来管理不同来源的联系人。
- 通过 Paypal (贝宝) 服务在应用程序中接受付款。

7. 开始入门

您可以了解在 Qt 应用程序框架中集成了 WebKit 浏览器引擎是如何为开发团队提供了两者的精华所在。这样，设计人员可使用熟悉的网络技术，其中包括 HTML、CSS 和 JavaScript；而编码人员可专心实现应用程序功能。而且整个团队可将本地功能扩展至网络环境，也可将网络控件集成在本地环境中，完全不受两种环境的限制。

- 要了解更多信息，请访问 <http://bit.ly/RJltf>
- 要查看更多操作示例，请查看 <http://bit.ly/Bt6Co>
- 然后通过 <http://bit.ly/17Ca8j> 下载 Qt SDK，包括 Qt Creator 和 Qt 4.5

关于 Qt Development Frameworks

Qt Development Frameworks (前身为 Trolltech) 是跨平台应用程序框架的全球领先者。诺基亚于 2008 年 6 月收购了 Trolltech, 并将其作为诺基亚内部的一个部门重新命名为 Qt Development Frameworks。借助 Qt, 开源和商业用户不用过多编写代码, 却可创建更多应用并可随时随地进行开发。开发人员使用 Qt 可一次性构建创新服务和应用程序, 无需重新编写代码就可跨所有主要桌面系统、移动和其他嵌入式平台部署全新开发。众多领先的消费性电子厂商也使用 Qt 创建用于 Linux 设备的高级用户界面。Qt 为诺基亚开发一系列用户倾慕的产品和体验的战略提供了稳固的基础。

关于诺基亚

诺基亚是全球最大的移动终端制造商, 引领互联网及通信融合产业的发展。诺基亚为广大的消费群体提供丰富的终端产品, 并通过互联网服务为人们带来音乐、地图、媒体、讯息和游戏体验。此外, 诺基亚还通过 NAVTEQ 提供丰富的电子地图信息, 通过诺基亚西门子通信公司为通信网络提供设备、解决方案和服务。

**NOKIA**