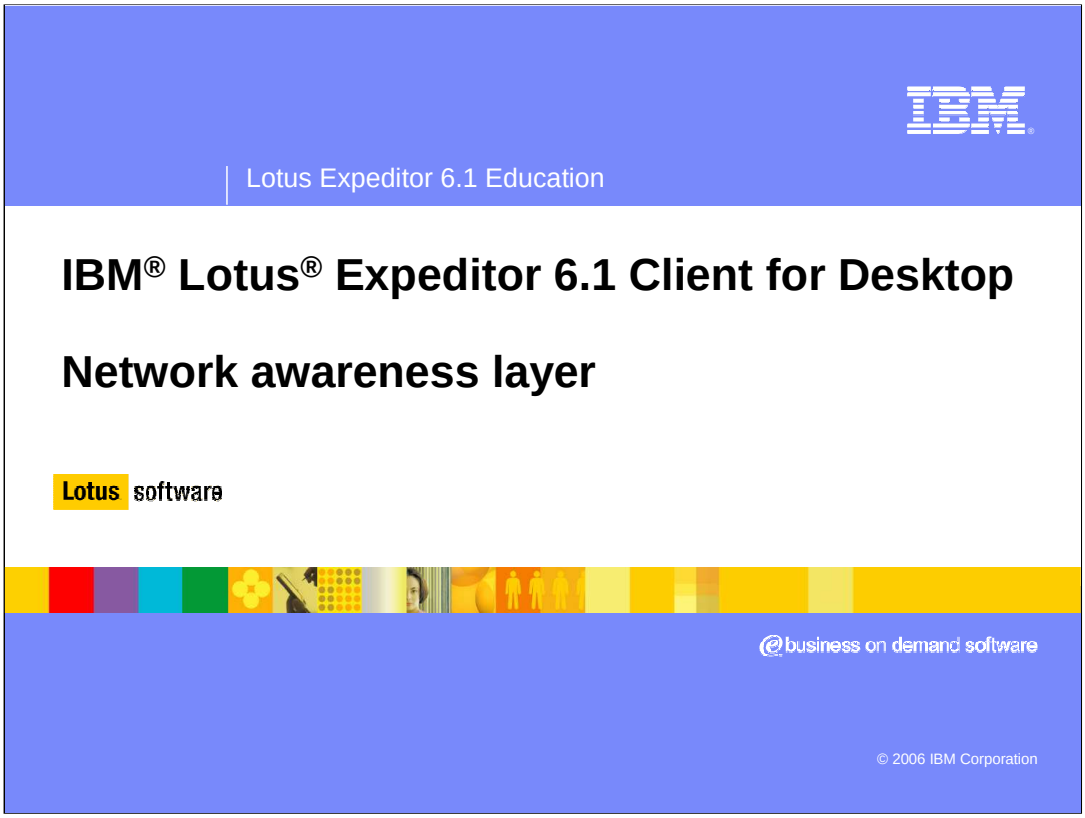


This presentation explains the Network Awareness Layer in IBM Lotus Expeditor 6.1 Client for Desktop.

Goals

- Understand the network awareness layer support provided in IBM Lotus Expeditor Client for Desktop

The goal of this presentation is to understand the network awareness layer support provided in IBM Lotus Expeditor 6.1 Client for Desktop.

Agenda

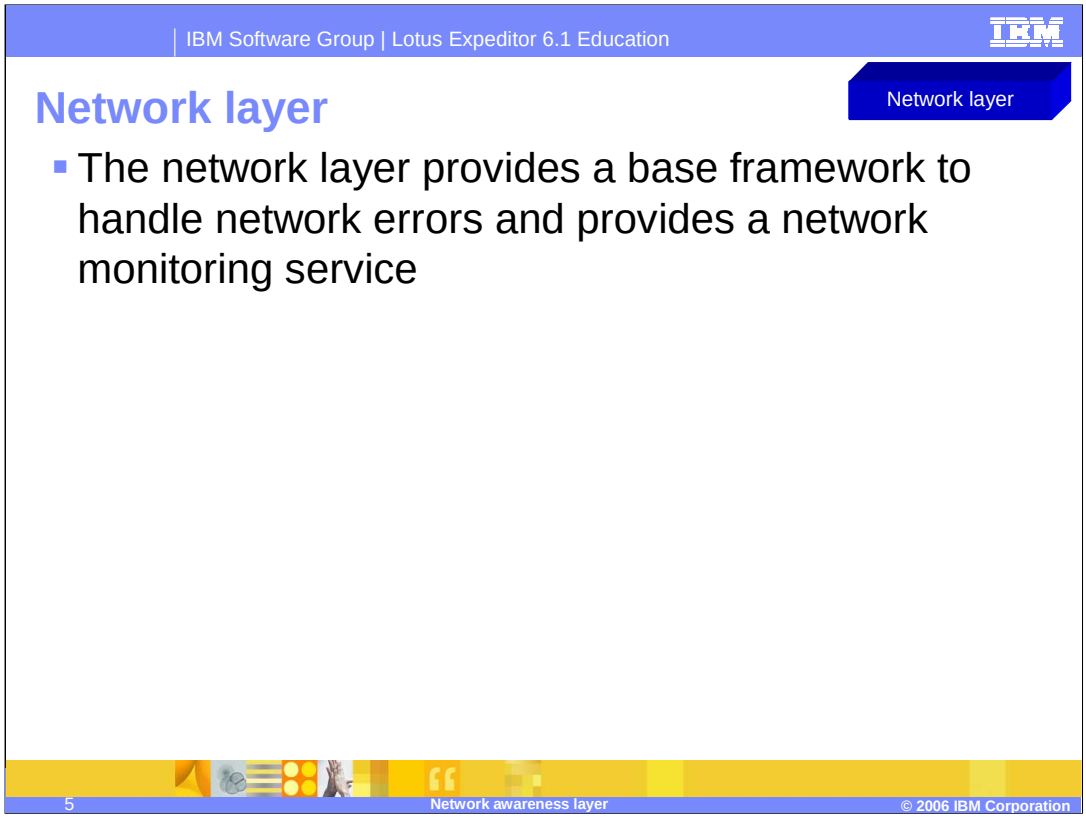
- Network Layer
  - Overview
  - Extension point
  - Notification Model
  - Customized Handlers
  - Configuration
  - RCP HTTP URL handler
  - Serviceability
  - Summary

The agenda of this presentation is to explain the Network Layer capabilities that the client platform provides to you, the infrastructure and plug-ins that enable these capabilities, and details about the network layer supported by IBM Lotus Expeditor 6.1 Client for Desktop.

Section

Network layer

So, let’s describe more details about the Network Layer



The network layer is a base network management layer in Expeditor Client that provides a framework to applications and platform components for network invocations and network error handling.

It provides users with the capability to determine the current status of the client platform as well as their connectivity to remote servers and HTTP resources including Web services. The application can choose to be notified of client status changes or check the client platform status by using the public APIs for offline manager. The application can handle the network error by using the public APIs for Net Faults. The application can also check the real network status by using the public APIs for network status. This layer also includes the Rich Client Platform (or RCP) version of HTTP URL Handler, which integrates with the account API and the offline manager.

Network layer functionality

Network layer

- Current network status of the client platform (online/offline)
- Connectivity status to remote servers
- Connectivity status to HTTP resources including Web services
- Application can be notified of client status changes
- Application can check the client platform status
  - Use the public APIs for Offline Manager
- Application can handle the network error
  - Use the public APIs for Net Faults
- Application can check the network adapter status
  - Use the public APIs for Network Status

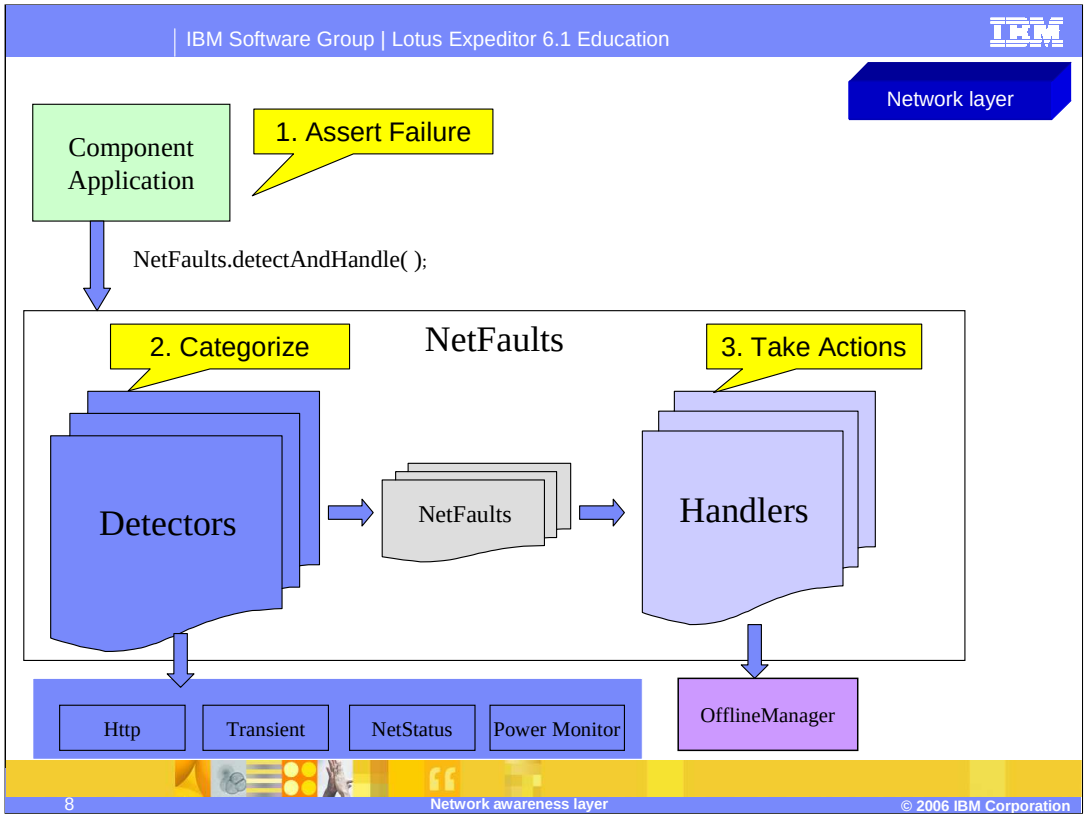
This slide describes the Network Layer functionality.

Network layer – Notification model

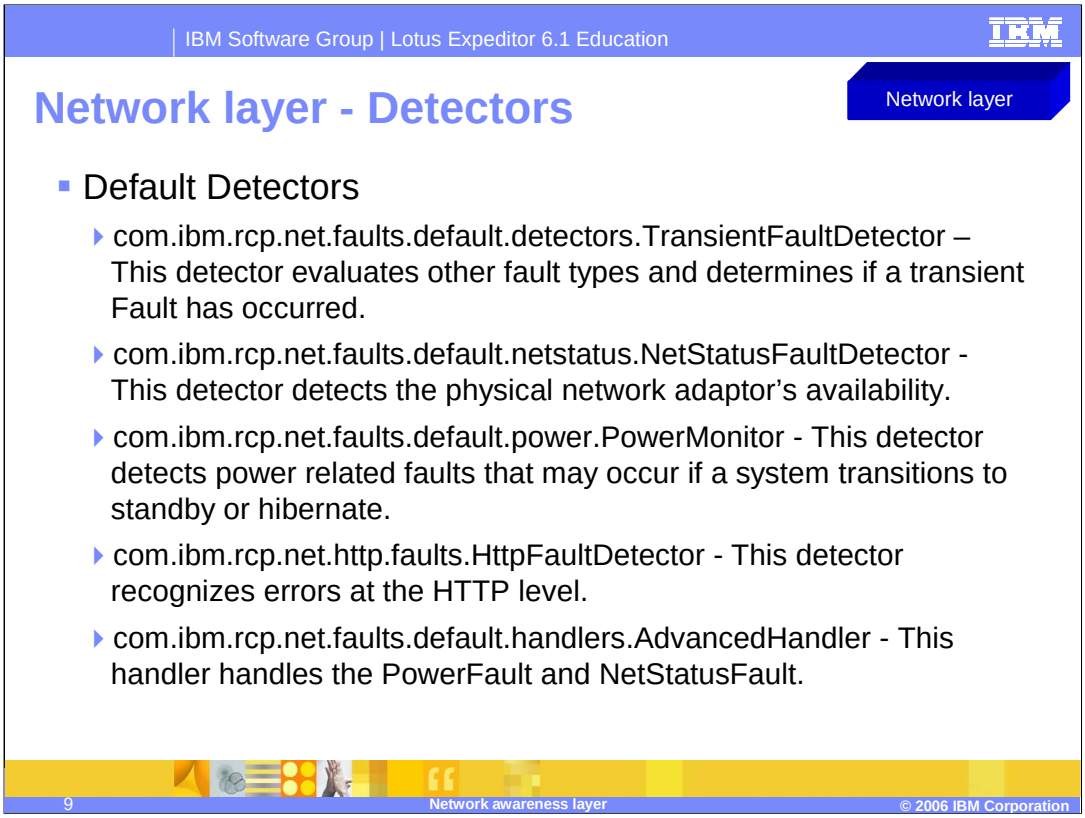
Network layer

- Applications use the Notification Model to be notified when
  - Platform network connection status change (online/offline)
  - Listening to remote server status change
- When an error occurs the application calls
  - com.ibm.rcp.net.faults.DetectAndHandle.detectAndHandle()
- DetectAndHandle()
  - Walk through the detectors, which generate the Faults
  - The Faults are passed to the handlers
  - The application component can also contribute customized handlers
- Handlers can then perform actions based on the Fault type

When an application throws a network error and calls NetFaults.detectAndHandle(), the NetFaults component will walk through the detectors and handlers in the order specified by the default configuration or its customized configuration. The detectors will categorize the fault and pass the fault to the handler. The handler can then take action based on the fault type. The platform provides all the detectors in a predefined order. The application component can contribute its customized handlers.



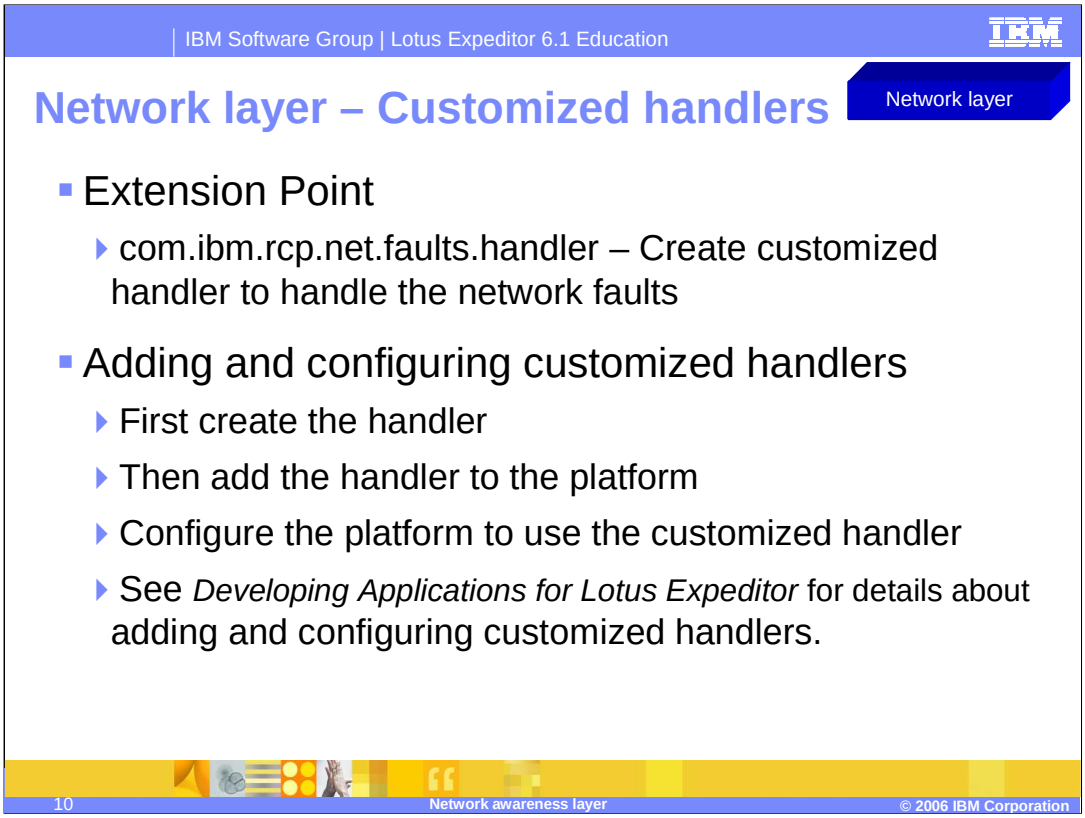
When a failure occurs, detectAndHandle is called by an application. The fault is categorized by the detectors and then passed to the Handlers where various actions can be done based on the fault occurring.



The Network Layer provides a handler extension point for applications to provide their own custom network faults handler.

The Network Layer also provides the following default detectors:

- The TransientFaultDetector evaluates other fault types and determines if a transient Fault has occurred.
- The NetStatusFaultDetector detects the physical network adaptor's availability.
- The PowerMonitor detector detects power related faults that may occur if a system transitions to standby or hibernate.
- The HttpFaultDetector recognizes errors at the HTTP level.
- The AdvancedHandler handles PowerFaults and NetStatusFaults



You can create customized handlers and configure the platform to use the customized handlers. First, you must create the handler. Then add the handler to the platform. See *Developing Applications for Lotus Expeditor* for information about creating a handler and adding the extension to the plugin.xml.

Network layer – Errors

- If you have a networking error, the NetStatusFaultDetector will generate a NetStatusFault by the NetStatusDetector
- When you use the HTTP plug-in, if the server is down or if there are networking errors, use one of these exception types:
  - SocketException
  - ConnectException
  - NoRouteToHostException
- Will generate a ConnectFault by the HTTPDetector

Let's discuss an example of Handling Faults. If you have a networking error, the NetStatusFaultDetector will generate a NetStatusFault by the NetStatusDetector. When you use the HTTP plug-in, if the server is down or if there are networking errors, the exception type SocketException, ConnectException, or NoRouteToHostException will generate a ConnectFault by the httpDetector.

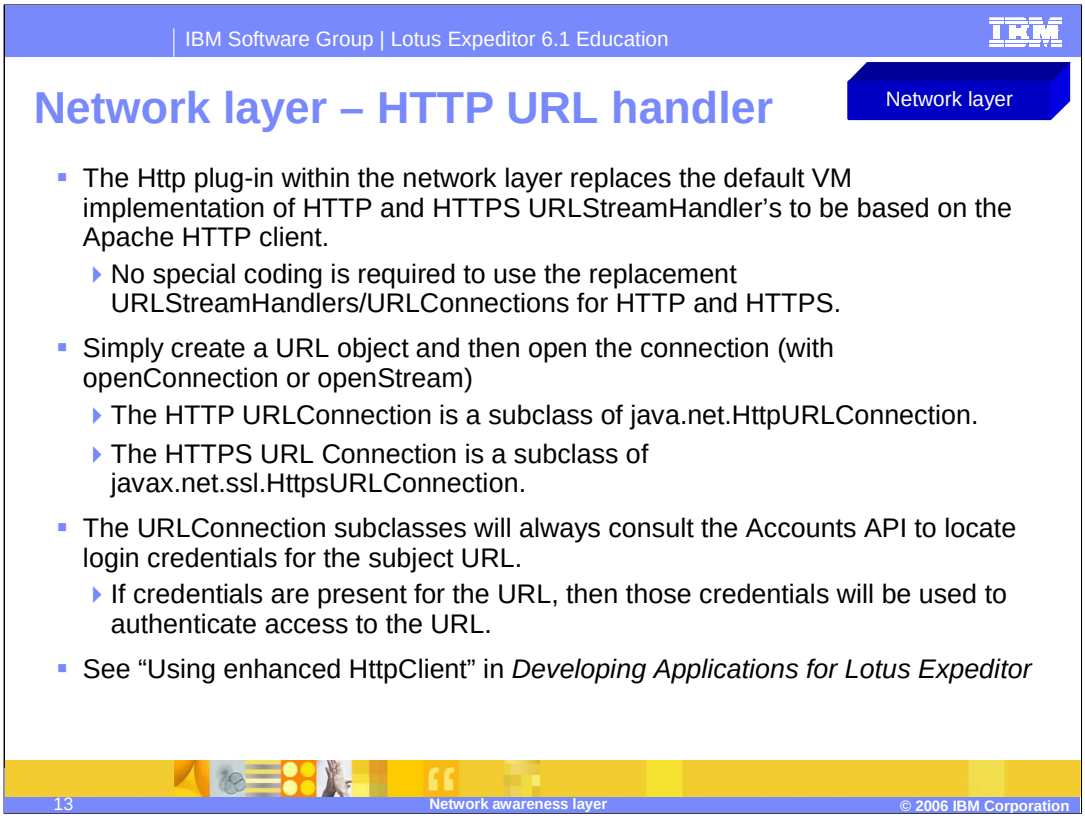
Network layer – Check network status

- Applications can use the NetStatus API to detect if the network is on or off.
  - The following is an example of how to use the API
    - hasAnyLinkConnecton() – the system has any link level connection
    - hasChanged() – indicates the network adapter has changed
    - update() – update to get the latest status of the network adapter

```
//call update() to get latest status, before checking status
private static final NetStatusMonitor _monitor=new NetStatusMonitor();
_monitor.update();
If (!_monitor.hasAnyLinkConnection())
   {
        //network is off
    }
```


Applications can use the Netstatus API to detect if the network is on or off.

This slide provides an example of how to use the API. Ensure you call the update() method first to get the latest status.



The network layer also includes the RCP version of HTTP URL Handler which integrates with the account API and the offline manager.

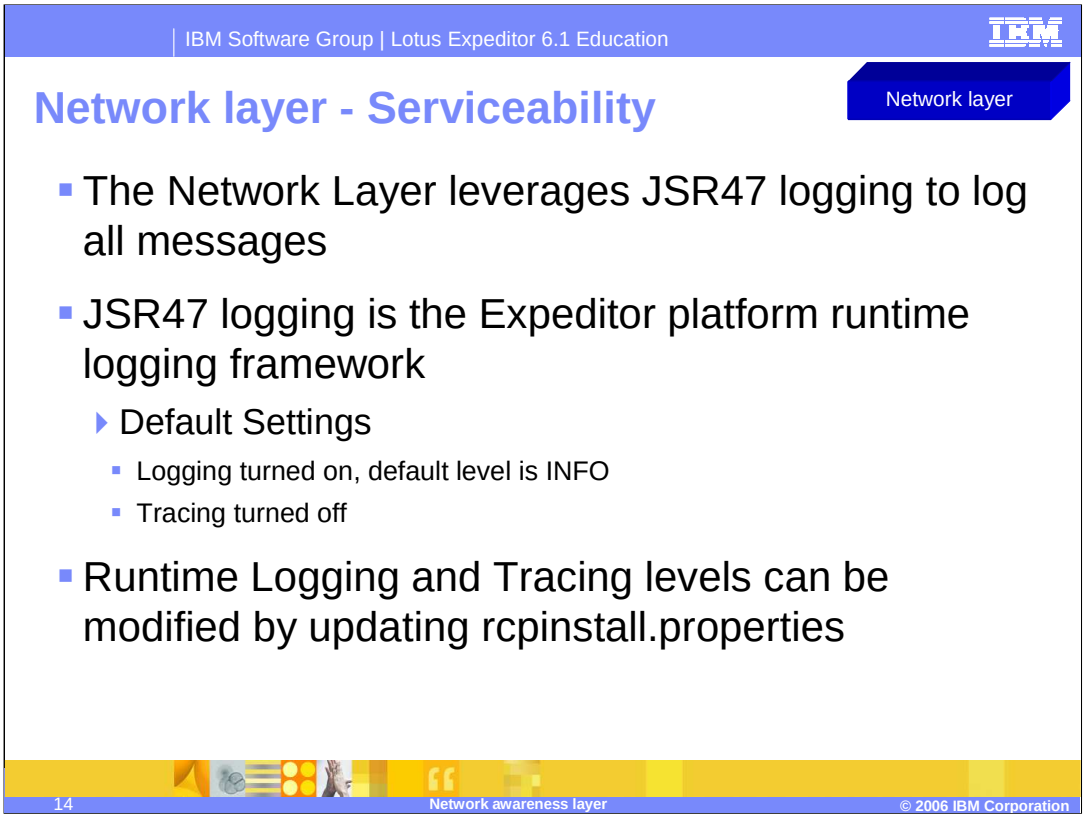
The Http plug-in within the network layer replaces the default VM implementation of HTTP and HTTPS URLStreamHandler's to be based on the Apache HTTP client.

No special coding is required to use the replacement URLStreamHandlers and URLConnections for HTTP and HTTPS. Simply create a URL object and then open the connection (with openConnection or openStream)

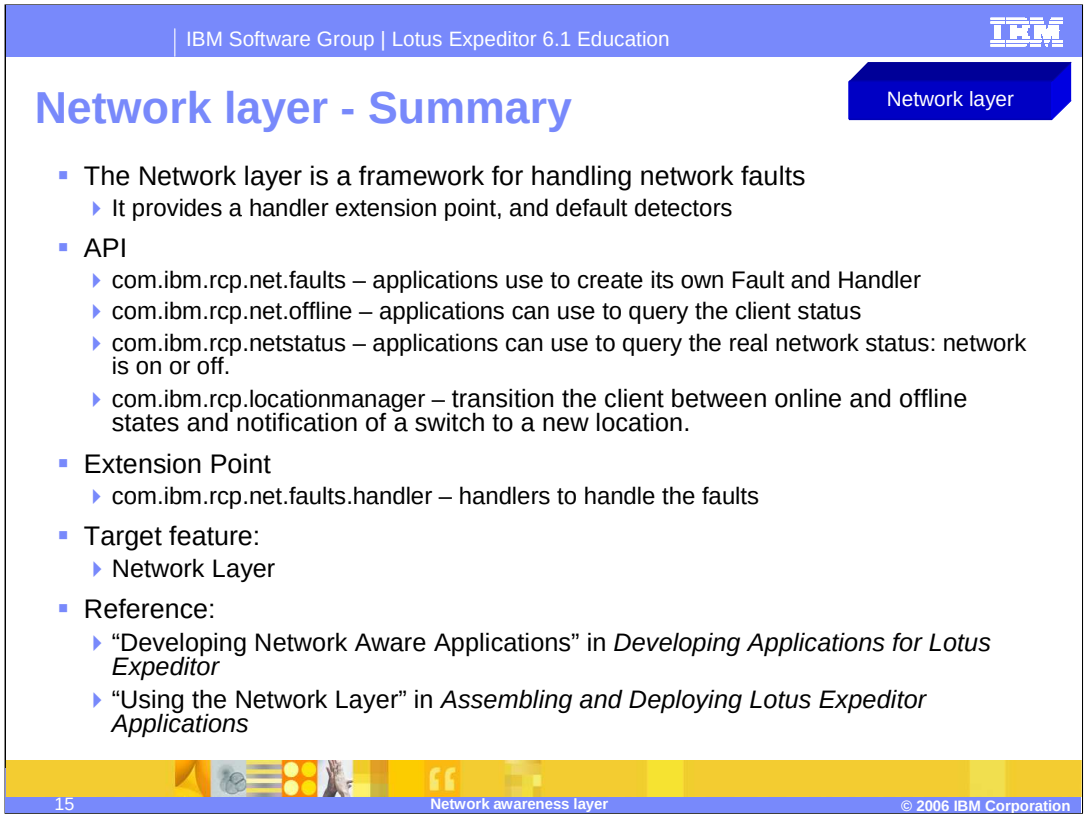
The HTTP URLConnection is a subclass of java.net.HttpURLConnection.

The HTTPS URL Connection is a subclass of javax.net.ssl.HttpsURLConnection but the methods introduced by javax.net.ssl.HttpsURLConnection are not implemented. Otherwise, the HTTPS URLConnection subclass operates normally.

The URLConnection subclasses will always consult the Accounts API to locate login credentials for the subject URL. If credentials are present for the URL, then those credentials will be used to authenticate access to the URL.



The Network Layer has logging and tracing facilities. The Network Layer leverages the JSR47 logging framework to log all messages. This is the Expeditor Client platform runtime logging framework. The default configuration of the platform logging framework is stored in the user's workspace/.config/rcpinstall.properties file. This is further documented in the guide *Developing Applications for Lotus Expeditor*.



In summary, the Network Layer provides a framework for handling network faults. Further documentation on the API and Extension point can be found in *Developing Applications for Lotus Expeditor* in the section “Developing Network Aware Applications” and in *Assembling and Deploying Lotus Expeditor Applications* in the section “Using the Network Layer”.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM Lotus

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

This concludes the presentation