



LongReach

Document Library

LongReach

Administrator Guide

Contents summary

WELCOME TO LONGREACH FOR ADMINISTRATORS	8
SECTION 1 – LONGREACH APP	9
WHAT IS LONGREACH?	9
USING THE LONGREACH APP WITH A SERVER	10
SECTION 2 – LONGREACH SERVER	14
ARCHITECTURE	14
FOLDERS	14
LONGREACH SERVER CONFIGURATION	17
USER ACCESS TO A LONGREACH SERVER.....	18
MOBILE DEVICE ACCESS TO A LONGREACH SERVER	19
MANAGE USER ACCESS TO FILES AND FOLDERS ON A SERVER	20
FILE TRANSFER NOTIFICATIONS	23
LOCATION DATA.....	25
DEPLOYMENT OPTIONS.....	25
SECTION 3 - ADMINISTRATION	30
ADMINISTRATOR RESPONSIBILITIES	30
DECISIONS ADMINISTRATORS NEED TO MAKE	30
PREREQUISITES	31
INSTALL LONGREACH SERVER COMPONENTS	31
CONFIGURATION FILES AND THEIR LOCATIONS	32
WHAT TO CONFIGURE FOR LONGREACH	32
USING HTTP OR HTTPS WITH LONGREACH	35
MANAGE LOG AND ERROR FILES	36
COLLECT DATA TO ASSIST IN FINDING ERROR CAUSES.....	37
TROUBLESHOOTING	39
GET SUPPORT FROM THE LONGREACH FORUMS	41
SECTION 4 - CONFIGURATION	42
THE CONFIGURATION PROCESS.....	42
ABOUT THE CONFIGURATION FILES.....	42
PORTS.....	44
JSM HTTP SERVER	45
ACCESS LOG	54
USER ACCESS PERMISSIONS	55
FOLDER ACCESS PERMISSIONS	58
FILE TRANSFER NOTIFICATIONS	62
CONFIGURE MULTIPLE INSTANCES AND VIRTUAL SERVERS	64
LONGREACH SERVER CONFIGURATION REFERENCE GUIDE	66
APPENDICES	75
ABBREVIATIONS AND TERMS	75
ASSUMED AND PREREQUISITE KNOWLEDGE	76
MOBILE DEVICE REQUIREMENTS.....	76
WHAT PORTS DOES LONGREACH USE?	77
DATA PROTECTION IN THE LONGREACH APP	78
USING EBCDIC FILES IN THE LONGREACH APP.....	78
FILE AND FOLDER ACCESS PERMISSIONS.....	79

Contents

WELCOME TO LONGREACH FOR ADMINISTRATORS	8
SECTION 1 – LONGREACH APP	9
WHAT IS LONGREACH?	9
<i>Install the LongReach app on a mobile device</i>	<i>10</i>
USING THE LONGREACH APP WITH A SERVER	10
<i>Turn communications on/off.....</i>	<i>10</i>
<i>Turn sounds on/off.....</i>	<i>10</i>
<i>Server list.....</i>	<i>10</i>
<i>Communications settings.....</i>	<i>11</i>
<i>LongReach app and server version compatibility.....</i>	<i>12</i>
<i>Using multiple user profiles on one server</i>	<i>13</i>
<i>Accessing multiple LongReach servers</i>	<i>13</i>
<i>Copy files and folders between LongReach servers.....</i>	<i>13</i>
SECTION 2 – LONGREACH SERVER	14
ARCHITECTURE	14
FOLDERS	14
<i>How the different folder types operate</i>	<i>15</i>
GroupBox	15
SharedBox	15
SubscribeBox	16
User folders.....	16
<i>Methods for sharing files and folders</i>	<i>16</i>
GroupBox	16
SharedBox	16
SubscribeBox	17
Share files and folders using group profiles	17
Share files and folders using object links	17
LONGREACH SERVER CONFIGURATION	17
<i>Configuration</i>	<i>17</i>
<i>File and folder locations</i>	<i>18</i>
USER ACCESS TO A LONGREACH SERVER.....	18
<i>User.allow and user.deny directives.....</i>	<i>18</i>
<i>How LongReach processes user allow and deny directives</i>	<i>18</i>
MOBILE DEVICE ACCESS TO A LONGREACH SERVER	19
<i>Device identification.....</i>	<i>19</i>
<i>Control access by mobile devices</i>	<i>19</i>
<i>Change the maximum device count</i>	<i>20</i>
<i>Remove named devices.....</i>	<i>20</i>
MANAGE USER ACCESS TO FILES AND FOLDERS ON A SERVER	20
<i>Controlling access to data on the LongReach server.....</i>	<i>20</i>
<i>Fully qualified names of user and shared folders</i>	<i>21</i>
SharedBox	21
User folders.....	21
<i>Folder get, put and delete directives.....</i>	<i>21</i>
Definitions.....	21
Folder access permissions.....	21
<i>Examples of folder get, put and delete directives</i>	<i>22</i>
Allow receive, send and delete actions in user and shared folders	22
Allow receive and send but not delete actions in the shared folder.....	22
Allow only receive access in a specific folder	23
FILE TRANSFER NOTIFICATIONS	23

<i>Data queues and notify server directives</i>	23
<i>Message content and format</i>	23
<i>Message oriented architecture and data queues</i>	24
LOCATION DATA	25
DEPLOYMENT OPTIONS	25
<i>LongReach with files and folders on one server</i>	25
<i>LongReach on multiple servers</i>	26
<i>LongReach on one server accessing remote servers</i>	26
<i>Multiple virtual servers</i>	27
<i>Configuring remote and virtual servers</i>	29
Host directives	29
Remote activation key	29
Prerequisite software	29
Accessing files and folders on remote servers	29
SECTION 3 - ADMINISTRATION	30
ADMINISTRATOR RESPONSIBILITIES	30
DECISIONS ADMINISTRATORS NEED TO MAKE	30
<i>Port numbers</i>	30
<i>Device count</i>	30
<i>Folder structure and naming conventions</i>	30
<i>File transfer notifications</i>	31
PREREQUISITES	31
INSTALL LONGREACH SERVER COMPONENTS	31
<i>Installation and configuration process</i>	31
<i>New installations</i>	32
<i>Upgrades</i>	32
CONFIGURATION FILES AND THEIR LOCATIONS	32
WHAT TO CONFIGURE FOR LONGREACH	32
<i>JSM HTTP Server</i>	32
<i>LongReach services</i>	34
User access permissions	34
Device access permissions	34
Folder access permissions	34
USING HTTP OR HTTPS WITH LONGREACH	35
<i>Public/private keys and certificates</i>	35
<i>Configure LongReach for SSL</i>	36
MANAGE LOG AND ERROR FILES	36
<i>File names and locations</i>	36
<i>Enable and disable logging</i>	36
<i>Archive log files</i>	37
<i>Delete log files</i>	37
COLLECT DATA TO ASSIST IN FINDING ERROR CAUSES	37
<i>Trace files: names and locations</i>	37
<i>Enable and disable tracing</i>	38
<i>Clear trace files</i>	38
<i>Archive trace files</i>	39
TROUBLESHOOTING	39
Error messages	39
Installation and configuration	40
GET SUPPORT FROM THE LONGREACH FORUMS	41
SECTION 4 - CONFIGURATION	42
THE CONFIGURATION PROCESS	42
ABOUT THE CONFIGURATION FILES	42

<i>manager.properties</i>	42
<i>httpd configuration file</i>	43
PORTS.....	44
JSM HTTP SERVER	45
<i>Server instance configuration</i>	45
<i>Controlling access to the server instance</i>	45
<i>MIME types for the server instance</i>	47
<i>Virtual host configuration</i>	48
Virtual host access	49
Virtual host protect	51
Virtual host script	52
Virtual host MIME types	54
ACCESS LOG	54
USER ACCESS PERMISSIONS	55
FOLDER ACCESS PERMISSIONS	58
<i>Base</i>	58
<i>Shared</i>	58
<i>Folder get, put and delete</i>	58
FILE TRANSFER NOTIFICATIONS	62
CONFIGURE MULTIPLE INSTANCES AND VIRTUAL SERVERS	64
<i>Anatomy of a configuration file</i>	64
<i>Multiple HTTP server instances</i>	65
<i>Multiple virtual servers</i>	65
LONGREACH SERVER CONFIGURATION REFERENCE GUIDE	66
<i>Configuration item reference</i>	66
<i>About MIME types</i>	70
<i>Address allow and deny directives syntax</i>	72
APPENDICES	75
ABBREVIATIONS AND TERMS	75
ASSUMED AND PREREQUISITE KNOWLEDGE	76
MOBILE DEVICE REQUIREMENTS	76
<i>Hardware and operating system</i>	76
<i>Software</i>	77
<i>Connectivity</i>	77
WHAT PORTS DOES LONGREACH USE?	77
DATA PROTECTION IN THE LONGREACH APP	78
<i>How LongReach data protection works</i>	78
<i>Apple iOS security hardening checklist</i>	78
USING EBCDIC FILES IN THE LONGREACH APP	78
FILE AND FOLDER ACCESS PERMISSIONS	79

Document revision date: Friday, 25 May 2012

List of Figures

FIGURE 1: LONGREACH APP SCREEN SAMPLES	9
FIGURE 2: COMMUNICATIONS TURN ON/OFF	10
FIGURE 3: COMMUNICATIONS SOUNDS TURN ON/OFF	10
FIGURE 4: SERVER LIST.....	11
FIGURE 5: COMMUNICATIONS SETTINGS FOR A SERVER	12
FIGURE 6: LONGREACH ARCHITECTURE AND COMPONENTS	14
FIGURE 7: LONGREACH AND FILE AND FOLDERS ON ONE SERVER.....	25
FIGURE 8: LONGREACH ON MULTIPLE SERVERS	26
FIGURE 9: LONGREACH ON ONE SERVER WITH FILES AND FOLDERS ON BOTH SERVERS.....	26
FIGURE 10: LONGREACH ON ONE SERVER WITH FILES AND FOLDERS ONLY ON ANOTHER SERVER.....	27
FIGURE 11: MULTIPLE VIRTUAL SERVERS IN LONGREACH	27
FIGURE 12: CHICAGO VIRTUAL SERVER ACCESS PRIVILEGES.....	28
FIGURE 13: HUMAN RESOURCES VIRTUAL SERVER ACCESS PRIVILEGES.....	28
FIGURE 14: SALES VIRTUAL SERVER ACCESS PRIVILEGES	28
FIGURE 15: HOW USERS ACCESS A PHYSICAL SERVER	29
FIGURE 16: LANSa INTEGRATOR PKI EDITOR.....	35
FIGURE 17: CONFIGURATION OF A SINGLE INSTANCE AND A SINGLE VIRTUAL SERVER.....	65
FIGURE 18: CONFIGURATION OF A SINGLE INSTANCE WITH MULTIPLE VIRTUAL SERVERS	65
FIGURE 19: CONFIGURATION OF MULTIPLE INSTANCES WITH MULTIPLE VIRTUAL SERVERS.....	65

List of Tables

TABLE 1: COMMUNICATIONS SETTINGS EXPLANATION	11
TABLE 2: USER ALLOW AND DENY DIRECTIVES PROCESSING LOGIC	18
TABLE 3: FOLDER GET, PUT AND DELETE DIRECTIVE DEFINITIONS.....	21
TABLE 4: DATA QUEUE CHARACTERISTICS.....	24
TABLE 5: CONFIGURATION FILE LOCATIONS.....	32
TABLE 6: WHAT TO CONFIGURE FOR THE SERVER INSTANCE	33
TABLE 7: ERROR MESSAGES	39
TABLE 8: EXPLANATION OF MANAGER.PROPERTIES	42
TABLE 9: EXAMPLE OF AN UPDATED MANAGER.PROPERTIES FILE	43
TABLE 10: STRUCTURE OF THE HTTPD CONFIGURATION FILE.....	43
TABLE 11: MANDATORY AND OPTIONAL CHANGES TO CONFIGURATION ITEMS AND PARAMETERS.....	44
TABLE 12: DEFAULT PORTS.....	44
TABLE 13: CONFIGURE A SERVER INSTANCE	45
TABLE 14: CONTROL ACCESS TO THE SERVER INSTANCE.....	46
TABLE 15: WHAT TO CONFIGURE FOR SERVER INSTANCE ACCESS	46
TABLE 16: MIME TYPES FOR THE SERVER INSTANCE	47
TABLE 17: WHAT TO CONFIGURE FOR THE SERVER INSTANCE MIME TYPES.....	48
TABLE 18: VIRTUAL HOST CONFIGURATION	48
TABLE 19: WHAT TO CONFIGURE FOR THE VIRTUAL HOST	48
TABLE 20: VIRTUAL HOST ACCESS DIRECTIVES.....	49
TABLE 21: WHAT TO CONFIGURE FOR VIRTUAL HOST ACCESS	50
TABLE 22: VIRTUAL HOST PROTECT CONFIGURATION	51
TABLE 23: CONFIGURING VIRTUAL HOST PROTECT	52
TABLE 24: VIRTUAL HOST SCRIPT CONFIGURATION	52
TABLE 25: VIRTUAL HOST MIME TYPE CONFIGURATION	54
TABLE 26: WHAT TO CONFIGURE FOR VIRTUAL HOST MIME TYPES.....	54
TABLE 27: USER ALLOW AND DENY PARAMETER SYNTAX	55
TABLE 28: EXAMPLE CONFIGURATION - USER ALLOW AND DENY PARAMETERS	55
TABLE 29: USER ALLOW AND DENY EXAMPLES.....	57
TABLE 30: FOLDER GET, PUT AND DELETE PARAMETER SYNTAX.....	58
TABLE 31: EXAMPLE CONFIGURATION – FOLDER GET PUT AND DELETE PARAMETERS	59
TABLE 32: FOLDER GET, PUT AND DELETE ALLOW AND DENY EXAMPLES – MANUALS FOLDER	61
TABLE 33: FOLDER GET, PUT AND DELETE ALLOW AND DENY EXAMPLES – BASE FOLDER	61
TABLE 34: FOLDER GET, PUT AND DELETE ALLOW AND DENY EXAMPLES – SHAREDBox FOLDER	61
TABLE 35: EXAMPLE CONFIGURATION – NOTIFY PARAMETER.....	62
TABLE 36: NOTIFY PARAMETER DEFINITION AND EXAMPLES.....	64
TABLE 37: SERVER REFERENCE: CONFIGURATION ITEM REFERENCE	66
TABLE 38: SERVER REFERENCE: MIME TYPES	70
TABLE 39: SERVER REFERENCE: ACCESS ALLOW AND DENY ADDRESSES.....	72
TABLE 40: SERVER REFERENCE: ACCESS ALLOW AND DENY CONTENT LENGTH.....	72
TABLE 41: SERVER REFERENCE: ACCESS ALLOW AND DENY USER AGENTS	73
TABLE 42: SERVER REFERENCE: SAMPLE LISTS OF USER AGENTS	74
TABLE 43: ABBREVIATIONS AND TERMS	75
TABLE 44: ASSUMED AND PREREQUISITE KNOWLEDGE.....	76

Welcome to LongReach for administrators

LongReach is a software application for managing files and folders on mobile devices and servers. It consists of two components, the LongReach app and the LongReach server.

The LongReach app is a native iOS application that provides file and folder management and file transfer between a server and mobile devices such as the iPhone and iPad. The application manages any type of file including, documents, presentations, spreadsheets, photographs, voice recordings, video, and text files. Files can be created on the mobile device or on the server and transferred securely between the two. The LongReach app works with the server components and also can operate as a standalone application on a mobile device.

This guide describes how to manage the server components of LongReach, administrator responsibilities and the configuration options available to administrators.

If you want to:	Refer to:
Know about the LongReach app	Section 1 – LongReach app Installing the LongReach app on a mobile device and configuring communications.
Understand what the LongReach server components do	Section 2 – LongReach server Learn about architecture, folders, users, sharing files, and access controls.
Learn out about administrator activities and responsibilities, fixing problems and managing day-to-day activities.	Section 3 – Administration This section explains administrative concepts, responsibilities, housekeeping and troubleshooting.
Set values for configuration items and parameters	Section 4 – Configuration This section explains how to configure LongReach Server by setting values for the configuration items and parameters. The section includes a reference guide for the JSM HTTP Server.
Read reference information	Appendices

Administrator Guide and LongReach versions

Use the version of the Administrator Guide that matches the installed version of the LongReach server software.



Use this copy of the LongReach Administrator Guide with LongReach server version 1.1

Section 1 – LongReach app

This section provides an overview of the LongReach mobile app describing how to install and configure the app. Refer to documentation supplied with the LongReach app for detailed information about using the app on a mobile device.

This section is an overview of how to install and configure LongReach on a mobile device.

What is LongReach?

The LongReach app is a native iOS application that provides file and folder management and file transfer between a server and mobile devices such as the iPhone and iPad. The application manages documents, presentations, spreadsheets, photographs, voice recordings, video, and text files. Files can be created on the mobile device or on the server and transferred securely between the two.

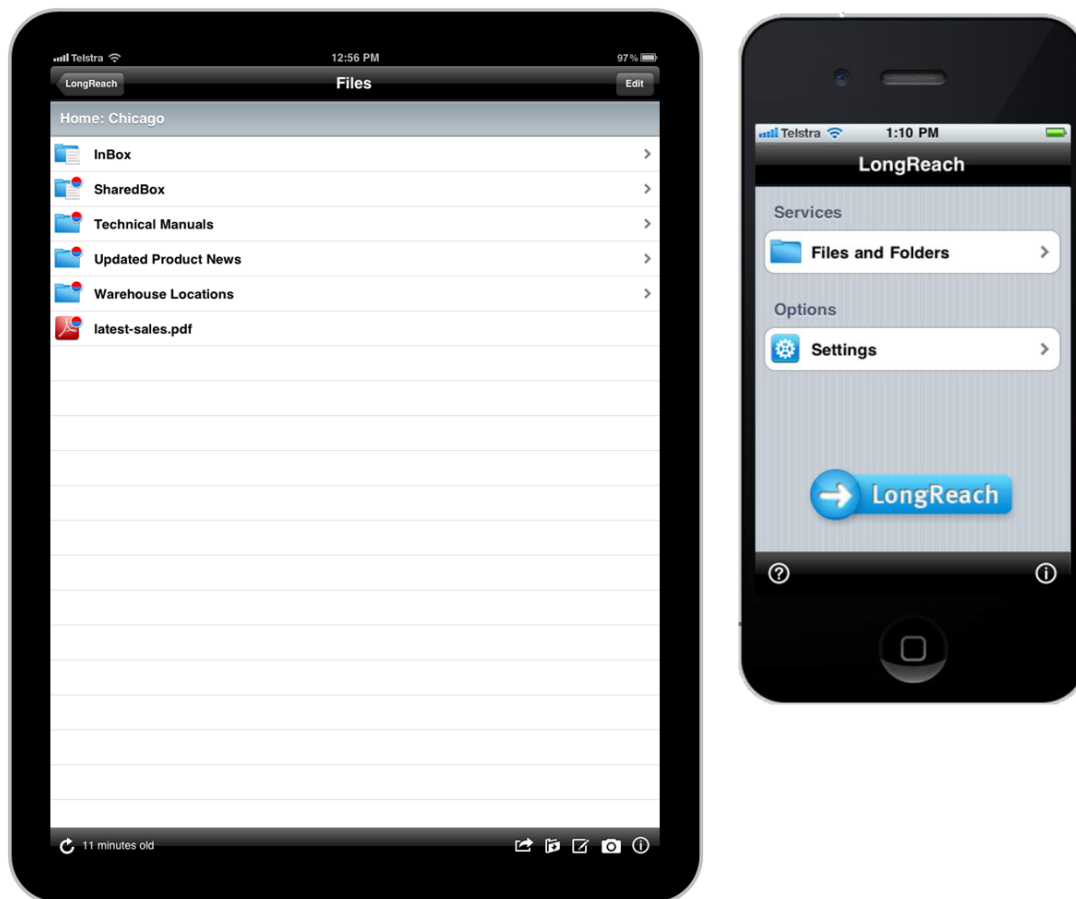


Figure 1: LongReach App Screen Samples

The mobile device application uses a colour-coded display that makes it easy to see files that are only on the server, files that are only on the mobile device or files that are on both the server and mobile device. When viewing a folder's content, it is obvious which files are local, and it's a simple task to choose local files to send to the server.

Files are encrypted while on the mobile device and during transmission to and from the server. Users must have an active user profile to use the server components. On IBM i servers, user permissions control the user's ability to create files and folders on the server.

Install the LongReach app on a mobile device

To use LongReach, download the app from the Apple Store (iTunes). No configuration is necessary to use LongReach as a stand-alone app on the mobile device.

To use the LongReach app with a LongReach server, install the app, then configure the servers and associated communications in the app settings.

Using the LongReach app with a server

The LongReach app can communicate with one or more LongReach servers. To enable this service turn on communications, turn on sounds (optional), allocate server names and configure the communications settings associated with each server.

Turn communications on/off

Communications must be turned on for the LongReach app to communicate with the LongReach server, and the server must be operating for the mobile device to connect successfully. Figure 2 (page 10) shows how to turn communications on and off.



Turn on		Allows the mobile device to communicate with the server.
Turn off		Prevents the mobile device from communicating with the server.

Figure 2: Communications Turn On/Off

Turning communications off only prevents the LongReach app from communicating with a LongReach server. This action does not turn off any of the other communications services on the mobile device.

Turn sounds on/off

The LongReach app can play sounds related to communications events. Playing the sounds is optional. Figure 3 (page 10) shows how to turn the sounds on and off.



Turn on		LongReach plays sounds for communications events.
Turn off		No sounds. LongReach does not play sounds for communications events.

Figure 3: Communications Sounds Turn On/Off

Server list

The server list is the starting point for configuring server communications settings, and provides space for up to five servers. Each server has its own collection of communications settings and storage location for files and folders.

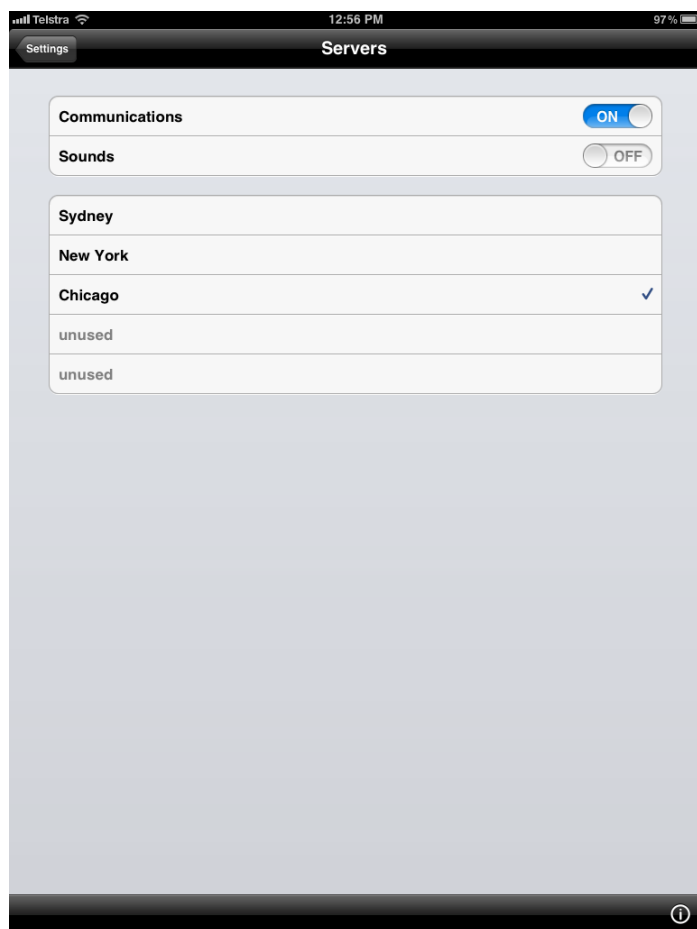


Figure 4: Server List

The server list in Figure 4 (page 11) shows three servers Chicago, New York and Sydney. A tick (✓) beside a server name indicates the active server and in this example Chicago is the active server. Only one server is active at any point in time.

Communications settings

Each server has a collection of communications settings that require configuration. Table 1 (page 11) describes the items in the communications settings.

Table 1: Communications Settings Explanation

Server	A name to identify the server.
Profile	A user name (or profile) known to the server.
Password	Password associated with the profile.
Host	DNS name or TCP/IP address of the server.
Port	TCP/IP port number to use when communicating with the server.
Path	URL for the LongReach service provided by the server.

HTTPS	<input checked="" type="checkbox"/> ON
	Use HTTPS protocol.
	<input type="checkbox"/> OFF
	Use HTTP protocol.

The LongReach app uses the server name to identify the server in the servers list, the communications settings and the files home page. The LongReach server does not use this name.

Figure 5 (page 12) illustrates the communications settings associated with the Chicago server.

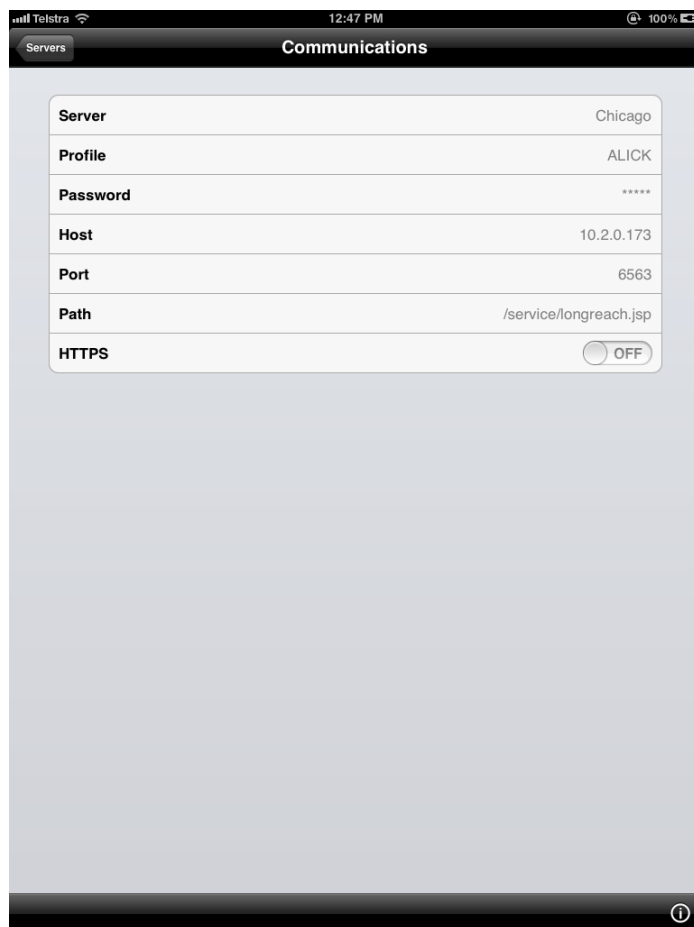


Figure 5: Communications Settings for a Server

Administrators are responsible for providing LongReach app users with values for profile, password, host, port, path and HTTP/HTTPS settings.

LongReach app and server version compatibility



LongReach will warn app users when their version of the app is incompatible with the version of the LongReach server. If this event occurs, upgrade either the app or the LongReach server software.

Using multiple user profiles on one server

LongReach users who have multiple user profiles on one server can configure a server and communications settings for each user profile in the LongReach app. Then they can switch between any of their user profiles without having to reconfigure user profile and server communications each time they want to use one of their user profiles.

For example, a person might have two profiles on the Chicago server, one as a member of the financial team and a second as an individual user. Configuring two servers in the LongReach app for the Chicago server allows the person to log on to the LongReach server using either profile.

Accessing multiple LongReach servers

A user may wish to communicate with multiple servers. The server and communications configurations allow users to configure and save multiple servers and associated user profile information. This feature makes it easy to switch between servers.

Users can switch from one server to another by changing only the active server. This eliminates the need to change communications configurations every time a user switches servers.

The LongReach server software must be installed on each server, or configured for access to remote servers.

Copy files and folders between LongReach servers

Copying files and folders from one LongReach server to another is a two step process. Step one, a LongReach app user connects to the server where the files and folders reside and transfers them from the server to the mobile device. Step two, connect to the LongReach server that is the intended destination of the files and folders, and then transfer the files and folders from the mobile device to the server.

Copying files and folders between LongReach servers is a two step process because file transfers are always from a server to the app, or from the app to a server. Copying files and folders is not a file transfer from one server directly to another server.

Section 2 – LongReach server

Read this section to discover conceptual information about the LongReach server.

Architecture

Figure 6 (page 14) illustrates the architecture of LongReach. The LongReach server is an application that provides file and folder management for server-based files and folders, and file transfer between a server and mobile devices.

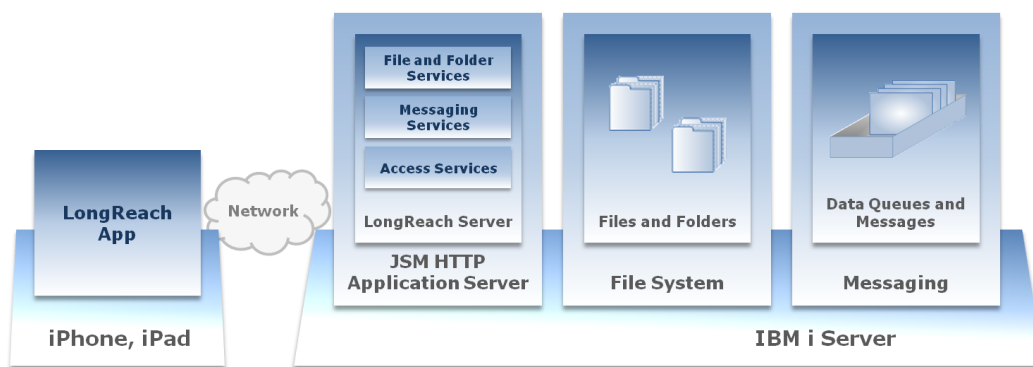


Figure 6: LongReach Architecture and Components

LongReach server components comprise an application server (JSM HTTP Server) and the LongReach server-based application. The IBM i Integrated File System (IFS) provides storage management services. LongReach server uses data queues to manage messages initiated by file transfers from the LongReach app.

Using the LongReach app installed on a mobile device, users can connect to the server and transfer (send or upload) files and folders to the server from their mobile device, or transfer (receive or download) files and folders from the server to their mobile device.

The LongReach app can place messages on a data queue at the successful completion of a send file transfer. The action to invoke this service in the LongReach app is send with notification. This optional feature simplifies integration of data collected on a mobile device with server-based applications and automated business processes.



The LongReach app can operate without the LongReach server.

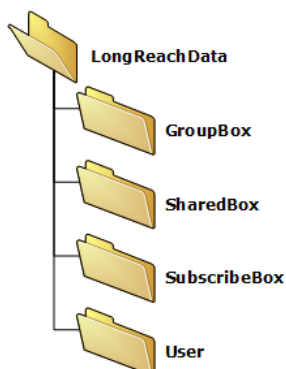
Folders

Folders are containers for files and sub-folders and reside on a mobile device, in the IFS on an IBM i server or in both places. This section explains the types of folders and describes how they operate in LongReach.

LongReach provides the following folder types:

- GroupBox
- InBox
- SharedBox
- SubscribeBox
- User folders

Each folder type has a set of properties and access policies that govern how the folders operate, how users access the folders, how users discover the folders, and the permissions to add or remove files and sub-folders.



GroupBox

GroupBox is a folder containing files and sub-folders created by a user and only accessible by users invited by the folder owner to join a group.

SharedBox

SharedBox is a public folder containing files and sub-folders accessible by all users.

SubscribeBox

SubscribeBox folders contain files and sub-folders published by a user and available to other users who wish to subscribe to the files and folders.

User folders

Each user has a folder on the server containing files and sub-folders belonging to the individual user.

InBox

InBox is a reserved folder that resides in the LongReach app on mobile devices. LongReach uses this folder to share or exchange files with other applications on the mobile device. Users can't delete the InBox folder.

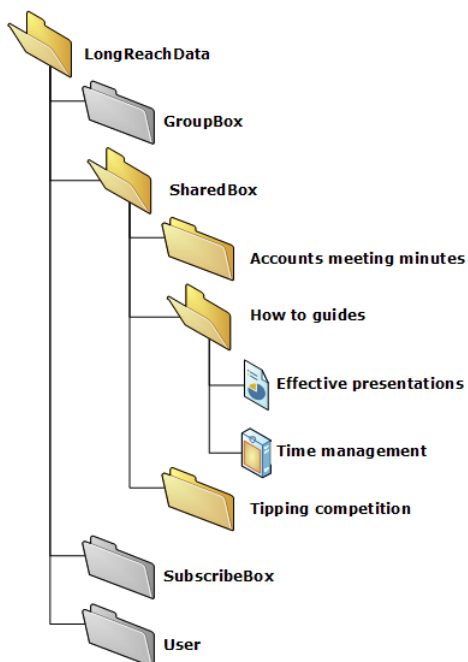
How the different folder types operate

GroupBox

This feature is coming in a later version of LongReach.

SharedBox

LongReach provides a shared folder that is available to all users.



SharedBox is a public folder where users may share files with other colleagues. The example shows files and folders in SharedBox.

Available actions are:

- Create sub-folders
- Delete sub-folders
- Delete files in folders and sub-folders
- Transfer files from folders and sub-folders
- Transfer files to folders and sub-folders

Any user can create folders and sub-folders in the SharedBox folder.

Authorised users operating on an IBM i server can access the SharedBox folder and its content from the server.

Programs operating on an IBM i server can access the SharedBox folder and its content.

SharedBox folders are available to any person operating the LongReach app that has an active user profile on an IBM i server. Administrators can use folder allow/deny directives in the LongReach server to prevent users adding, retrieving or deleting files and sub-folders in the SharedBox folder.

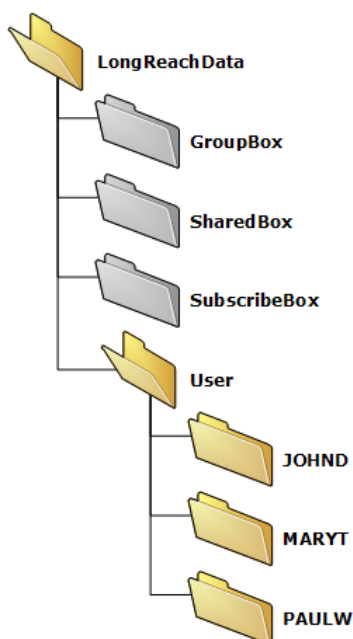
The LongReach server and the LongReach app will not allow users to delete the SharedBox folder on the server or a mobile device.

SubscribeBox

This feature is coming in a later version of LongReach.

User folders

LongReach creates a user folder for each person authorised to use a LongReach server.



User folders are for exclusive use by the person who owns the user folder. The example shows three user folders named JOHND, MARYT and PAULW.

Available actions are:

- Create sub-folders
- Delete sub-folders
- Delete files in folders and sub-folders
- Transfer files from folders and sub-folders
- Transfer files to folders and sub-folders
- Share user folders

Authorised users operating on an IBM i server can access user folders from the server.

Authorised programs operating on an IBM i server can access user folders.

Users can access files and sub-folders in their user folder but not in user folders belonging to other users.

Administrators can create folder sharing links to publish the files and sub-folders from a user folder for use by other users.

Methods for sharing files and folders

LongReach provides multiple ways to share files and folders using its special folder types GroupBox, SharedBox and SubscribeBox.

The object link features of the IBM i server provide other options for sharing files and folders. Administrators can use these features with LongReach.

GroupBox

This feature is coming in a later release of LongReach.

SharedBox

SharedBox is the simplest way for users to share files and folders. All LongReach app users will see the SharedBox folder in their list of folders on the server. They can transfer files from their mobile device to the SharedBox folder on the server or from the server to their mobile device.

SubscribeBox

This feature is coming in a later release of LongReach.

Share files and folders using group profiles

One way to share files and folders for a group of users is to create a group profile and allow group members to log on to the LongReach server using the group profile. Group members can access files and sub-folders in the group profile's user folder. In the LongReach app, group members will configure one server for their personal user profile and a second server for the group profile.

An example of sharing using a group profile is a project team that will work together for several months. The team can use a group profile to share documents associated with the project. The project files and sub-folders will reside under the group user profile. Only users who know the group profile can access and use the documents. Team members can work for different companies and share the documents. Disabling the group profile at the conclusion of the project will ensure protection against unauthorised access to files and folders.

This sharing method requires an administrator with user profile authority to create the group profile. GroupBox provides an alternative way to share files and folders among a group of users; and users can create groups and invite members without the assistance of an administrator.

Share files and folders using object links

On IBM i servers, object links to files and folders provide a way to share files and folders with multiple users. The add link command (ADDLNK) creates an object containing a pointer to the real object.

For example, suppose the user JOHND creates two documents to share with MARYT and PAULW. He saves Assumptions.doc and QuarterlyFigures.doc in his budget papers folder. The fully qualified locations of the documents are as follows.

Assumptions.doc: longreachdata/user/JOHND/Budget Papers/Assumptions.doc

QuarterlyFigures.doc is: longreachdata/user/JOHND/Budget Papers/QuarterlyFigures.doc

To share the documents with MARYT, JOHND creates a link using this ADDLNK command:

```
ADDLNK OBJ('/longreachdata/user/JOHND/Budget Papers')
NEWLNK('/longreachdata/user/MARYT/Accounts') LNKTYPE(*SYMBOLIC)
```

MARYT will see the documents in JOHND's Budget Papers folders in her Accounts folder.

To share the documents with PAULW, JOHND creates a link using this ADDLNK command:

```
ADDLNK OBJ('/longreachdata/user/JOHND/Budget Papers')
NEWLNK('/longreachdata/user/PAULW/Sales') LNKTYPE(*SYMBOLIC)
```

PAULW will see the documents in his Sales folder.

All three users will see the documents from the LongReach app on their mobile device in their respective folders.

This method for sharing files and folders relies on IBM i operating system features. Refer to documentation provided by IBM for more information about object links. The IBM i nomenclature for folder is directory.

LongReach server configuration

Configuration

The LongReach server is configurable and administrators control server activities by placing directive statements and parameters in the configuration file (httpd.xml). LongReach comes with a default configuration file. You do not need to change the default configuration if the default directives meet your processing, access control and security requirements.

Examples of directives are:

```
<parameter name="service.user.deny" value="JOHND"/>
```

```
<parameter name="service.folder.get.allow" value="{SHARED}"/>
```

Administrators can change the default configuration by adding directives, and/or changing default directives. The most likely configuration changes are allowing and denying permission to use the LongReach server and controlling access to files and folders.

File and folder locations

LongReach consists of files belonging to its server components and data files managed by LongReach. Both sets of files reside in folders in the server's file system, with data files separated from LongReach server files.

The default root folder for LongReach server files is /longreach.

The default root folder for data is /longreachdata.

User access to a LongReach server

Administrators can restrict access to a LongReach server by user. Exercise care when setting up access controls as it is possible to prevent everyone from using a LongReach server. Define an access control strategy based on both the sensitivity of the data in documents on the server and the users you will authorise access to the server.

User.allow and user.deny directives

Allow and deny directives in the LongReach server configuration define access permissions for individual users and/or user groups. Allow directives define users permitted to use the server and deny directives define users excluded from using the server. Examples of user.allow and user.deny directives are:

```
<parameter name="service.user.deny" value="JOHND"/>
```

```
<parameter name="service.user.allow" value="*USER"/>
```

These directives deny access to the user JOHND and allow access to all other users. The value "JOHND" is an explicit allow directive and the value "*USER" is a generic allow directive.

Insert user.allow and user.deny directives in the httpd.xml configuration file to implement user access control, using as many allow and deny directives as is necessary for your requirements. LongReach imposes no limit on the number of these directives.

LongReach does not support partial user profiles, examples are:

```
<parameter name="service.user.allow" value="M*/>"
```

```
<parameter name="service.user.deny" value="E*/>"
```

LongReach will ignore directives that define a value that is a partial name.

How LongReach processes user allow and deny directives

LongReach evaluates the deny directives first and then the allow directives. Table 2 (page 18) describes the processing logic.

Table 2: User Allow and Deny Directives Processing Logic

Step	Directives	Evaluation	Access Decision
1.	Deny	Any user profile with a user.deny directive will not receive permission to access the LongReach server.	Not permitted

Step	Directives	Evaluation	Access Decision
2.	Allow	Any user profile with a user.allow directive will receive access permission, including explicit and generic allow directives.	Permission granted
3.	Not covered by deny or allow	User profiles not included in deny or allow directives will not receive access permission.	Not permitted

The sequence of directives in the configuration file is of no consequence. However, the optimum sequence is the deny directives followed by the allow directives. This sequence groups directives so that it is easy for administrators to see users explicitly denied or allowed.

LongReach denies access to any user profile beginning with the letter Q. To override this access denial, each of these user profiles needs an explicit user allow directive.

Be careful when using `<parameter name="service.user.deny" value="*USER"/>` as this directive denies access to all users. You might use this directive to lock out all users temporarily.

To apply changes to service.user.deny and service.user.allow directives you must stop and restart the LongReach server.

Mobile device access to a LongReach server

Administrators can restrict access to a LongReach server by named mobile device.

Device identification

When the LongReach app runs the first time on a mobile device, it generates a device-identification and stores the identification securely in the application key store. All transactions between the LongReach app on a mobile device and a LongReach server include this device identification.

An example of a device-identification is A8627333-FD42-4EDC-B8AB-EA6A8A87BE13

Administrators can configure a LongReach server to allow or deny connections based on the device identification.

Uninstalling and reinstalling the LongReach app on a mobile device will generate a new device-identification. However, uninstalling and reinstalling LongReach app won't remove the original device-identification from the list of named devices residing on the server.

Control access by mobile devices

The LongReach server will accept connection requests from named mobile devices and a fixed number of mobile devices. A named mobile device is one that has a device-identification generated by LongReach. Administrators can control the mobile devices that may access a LongReach server using a combination of device count and named user devices.

When the LongReach server starts it reads a device access configuration file (device-longreach.txt) to determine the maximum number of devices allowed and the current list of named devices. The format of the configuration file is:

```
#LONGREACH,maximum-device-count,verification-hash

Device-identification,user

Device-identification,user
```

When the LongReach app on a mobile device starts for the first time LongReach generates a device-identification that is used for the life of the installed application (on that device).

An example of a device-longreach.txt file is:

```
#LONGREACH,500,1110FDADB87D3860384D0FD7D66430B5D81FB8D9  
2B5E88BA-3AEA-4F63-B06F-F13ED3C8BABD,JOHND
```

This example allows a maximum device count of five hundred (500) devices and includes one device, with device-identification 2B5E88BA-3AEA-4F63-B06F-F13ED3C8BABD, belonging to user JOHND.

The device-longreach.txt file will include a record for each device and LongReach will add devices automatically as devices connect for the first time. This process will continue until the device count reaches the maximum device count, and from then on no new devices will connect successfully.

The verification hash 1110FDADB87D3860384D0FD7D66430B5D81FB8D9 locks the file to the serial number of the server and the maximum number of devices.



The verification hash is not used in this version of LongReach server.

Change the maximum device count

Administrators can change the maximum device count at any time by editing the device access configuration file (device-longreach.txt) to increase or decrease the value of the maximum device count. Stop and restart the LongReach server for the changed value to take effect.

Remove named devices

Administrators can remove individual named devices by editing the device access configuration file (device-longreach.txt) and deleting records from the file. Stop and restart the LongReach server for the changed value to take effect.

Manage user access to files and folders on a server

The LongReach server provides folder directives that administrators can use to control user access to files and folders on a LongReach server. The folder directive manages receive (get), send (put) and delete actions. A receive (get) action occurs when a user transfers a file from a server to a mobile device. A send (put) action occurs when a user transfers a file from a mobile device to a server. Delete occurs when a user deletes a file or folder on the server. Folder directives apply to a specific folder or a folder and its sub-folders. Sub-folders inherit the directives from their parent folders, except when a sub-folder has specific folder directives.

On IBM i servers LongReach folder directives augment ownership and access control services IBM provides for the IFS. A user may be allowed to delete files from a folder based on IFS permissions, but the user will not be able to delete files using the LongReach app if the LongReach configuration includes specific delete deny directives.

Controlling access to data on the LongReach server

People using LongReach have access to corporate data. Corporate best practice suggests that companies ought to know and control who accesses corporate data in unstructured formats such as documents, images, presentations and spreadsheets.

Compliance and regulatory obligations require companies to maintain audit logs describing who accesses corporate information.

For these reasons administrators ought to allow only registered users to use LongReach server components. Each person who uses LongReach needs identifying credentials (unique user name/profile and password). Shared or generic user identifications will compromise your ability to determine who accesses the data.

Fully qualified names of user and shared folders

The fully qualified name of a folder defines the physical location of the folder. The LongReach configuration uses a virtual folder (for example `service.folder.shared`) as a variable whose contents represents the real folder.

SharedBox

The `service.folder.shared` directive defines the fully qualified name of the SharedBox folder:

```
<parameter name="service.folder.shared" value="/longreachdata/shared"/>
```

The folder directive uses this definition to substitute for `{SHARED}` in folder directives that define access policies for the SharedBox folder.

User folders

The `service.folder.base` directive defines the fully qualified name of user folders:

```
<parameter name="service.folder.base" value="/longreachdata/user/{NAME}"/>
```

LongReach substitutes the user profile name (upper case) for `{NAME}` and hence the fully qualified name of JOHND's user folder is: `/longreachdata/user/JOHND`

Use the lower case `{name}` to substitute the user profile name as lower case.

```
<parameter name="service.folder.base" value="/longreachdata/user/{name}"/>
```

The folder directive uses this definition to substitute for `{BASE}` in folder directives that define access policies for user folders.

Folder get, put and delete directives

Definitions

The folder get, put and delete directives control the actions available to users for accessing files and folders. Table 3 (page 21) provides definitions of the folder directives.

Table 3: Folder Get, Put and Delete Directive Definitions

Directives	Definitions
<code>service.folder.get.deny</code>	Users can't receive files and sub-folders
<code>service.folder.get.allow</code>	Users can receive files and sub-folders
<code>service.folder.put.deny</code>	Users can't send files and sub-folders
<code>service.folder.put.allow</code>	Users can send files and sub-folders
<code>service.folder.delete.deny</code>	Users can't delete files and sub-folders
<code>service.folder.delete.allow</code>	Users can delete files and sub-folders

Exercise care when defining folder directives to avoid permission conflicts. For example, LongReach folder directives may conflict with the server operating system access controls.

Folder access permissions

The LongReach server implements access permissions by inbuilt and explicit access controls. Administrators can implement explicit access permissions using the folder get, put and delete

directives. The LongReach server applies inbuilt access permissions to user folders. Administrators cannot override the inbuilt access controls.

Each user folder (/longreachdata/user/JOHND is an example) has get, put and delete access permissions assigned to the user who owns the folder. Administrators do not need to configure explicit access permissions for user folders.

The default access permissions for shared folders are allow get, put and delete for each users own folders.

Explicit access permissions are necessary when you want to restrict what users can do in certain folders.

Document editors can place files in the Product Information folder from the server (without using LongReach). They can't use the LongReach app to send files to the folder as the service.folder.put.deny prevents the LongReach app from sending files to the folder.

Examples of folder get, put and delete directives

Administrators can use multiple folder directives to control access to files and folders. This section provides examples of how to configure folder directives for specific access control requirements. The examples are indicative of real world contexts but may not reflect your precise requirements.

Allow receive, send and delete actions in user and shared folders

With this set of folder directives users can use receive (get), send (put) and delete actions in their user folder and sub-folders and the shared folder and its sub-folders.

```
<parameter name="service.folder.get.allow" value="{BASE}"/>
<parameter name="service.folder.get.allow" value="{BASE}/*/"/>
<parameter name="service.folder.put.allow" value="{BASE}"/>
<parameter name="service.folder.put.allow" value="{BASE}/*/"/>
<parameter name="service.folder.delete.allow" value="{BASE}"/>
<parameter name="service.folder.delete.allow" value="{BASE}/*/"/>
<parameter name="service.folder.get.allow" value="{SHARED}"/>
<parameter name="service.folder.get.allow" value="{SHARED}/*/"/>
<parameter name="service.folder.put.allow" value="{SHARED}"/>
<parameter name="service.folder.put.allow" value="{SHARED}/*/"/>
<parameter name="service.folder.delete.allow" value="{SHARED}"/>
<parameter name="service.folder.delete.allow" value="{SHARED}/*/"/>
```

This set of folder directives represents the least access control and allows any user to receive, send and delete files and folders.

Allow receive and send but not delete actions in the shared folder

This set of folder directives allows users to receive (get) and send (put) files and folders in the shared folder but no user can delete files and sub-folders in the shared folder.

```
<parameter name="service.folder.get.allow" value="{SHARED}"/>
<parameter name="service.folder.get.allow" value="{SHARED}/*/"/>
<parameter name="service.folder.put.allow" value="{SHARED}"/>
<parameter name="service.folder.put.allow" value="{SHARED}/*/"/>
<parameter name="service.folder.delete.deny" value="{SHARED}"/>
<parameter name="service.folder.delete.deny" value="{SHARED}/*/"/>
```

Allow only receive access in a specific folder

Suppose the shared folder contains a folder named Tutorials and it includes training materials that users may use but not change. These directives provide receive (get) access to files in the Tutorials folder but not send (put) or delete access.

```
<parameter name="service.folder.get.allow" value="{SHARED}/Tutorials"/>
```

```
<parameter name="service.folder.put.deny" value="{SHARED}/Tutorials"/>
```

```
<parameter name="service.folder.delete.deny" value="{SHARED}/Tutorials"/>
```

The put.deny directive prevents users from placing files in the Tutorials folder.

The delete.deny directive prevents users from removing files from the Tutorials folder.

File transfer notifications

The LongReach server can place messages on a data queue at the completion of a successful send file transfer from a mobile device. This service is useful when you want to notify an automated process on the server when specific files arrive. The automated process can retrieve the files and perform actions related to business activities. LongReach's responsibility is complete once the messages are on the queue. You need to write programs to monitor the queue, retrieve the messages and process the message content.

Data queues and notify server directives

Creating the resources to manage messages generated by file transfer notifications requires two steps:

Step 1: Create a data queue using server operating system commands.

On IBM i servers the command is CRTDTAQ.

Step 2: Insert a notify directive in the httpd.xml configuration file.

Refer to IBM documentation for information about creating data queues.

Examples of the LongReach notify directives are:

```
<notify data="{FOLDER}/{FILE}" key="{NAME}" padkey="true" ccsid="0"
queue="/QSYS.LIB/LONGREACH.LIB/NOTIFYUSER.DTAQ"/>
```

```
<notify data="{NAME},{FOLDER}/{FILE}" ccsid="0"
queue="/QSYS.LIB/LONGREACH.LIB/NOTIFYFILE.DTAQ"/>
```

Use a data queue name that indicates the purpose of the data queue.

Message content and format

The data parameter of the notify directive defines message content and format. Messages may contain any characters acceptable to data queues on a server. LongReach can insert variable data for folder name {FOLDER}, file name {FILE} and user profile {NAME}.

Example of a keyed data queue

This data queue is keyed by user profile {NAME} and the message content is the folder name, a forward slash (/) and the file name.

```
<notify data="{FOLDER}/{FILE}" key="{NAME}" padkey="true" ccsid="0"
queue="/QSYS.LIB/LONGREACH.LIB/NOTIFYUSER.DTAQ"/>
```

The message content is: folder name/file name

Example of a data queue without a key

This data queue has no key and the message content is the user profile, a comma, folder name, a forward slash and the file name.

```
<notify data="{NAME},{FOLDER}/{FILE}" ccsid="0"
queue="/QSYS.LIB/LONGREACH.LIB/NOTIFYUSER.DTAQ"/>
```

The message content is: user,folder name/file name

Example of extended message data

Messages in this data queue contain user profile, a comma, folder name, a forward slash, file name, a dash, text and a period.

```
<notify data="{NAME},{FOLDER}/{FILE}-Message created by LongReach." ccsid="0"
queue="/QSYS.LIB/LONGREACH.LIB/NOTIFYUSER.DTAQ"/>
```

The message content is: user,folder name/file name-Message created by LongReach.

Message oriented architecture and data queues

Data queues on IBM i servers are built on a messaging architecture and programs can exchange data without tightly coupled, program-to-program integration.

Table 4: Data Queue Characteristics

Characteristics	Why is this Useful?
Fast data exchange between programs and jobs	Data queues are an excellent way to synchronise and pass data between programs and jobs. Many programs and jobs can simultaneously access a data queue or queues.
Messages on a data queue are free format.	Messages can contain any data in any convenient format, thereby providing a flexible method for exchanging data.
Synchronous or asynchronous message processing	Data queues can be used for either synchronous or asynchronous processing. Programs receiving messages do not need to be active when the message is sent unless requirements dictate synchronous processing.
Priority message processing	Messages on data queues can be ordered in one the following ways: Last-in first-out (LIFO) - the last (or newest) message placed on the queue is the first message taken off the queue. First-in first-out (FIFO) - the first (or oldest) message on the queue is the first message taken off the queue. Keyed - messages on the queue have an associated key. Message can be taken off the queue by specifying the message key. This characteristic allows processing messages by a priority rather than just a fixed arrival sequence.

This feature of LongReach simplifies integration of data collected on mobile devices with applications on the server. LongReach transfers files and places messages on a queue. A server-based application can retrieve the messages and process data in the files.

Location data

When a user drops a pin onto a LongReach location map, the title and subtitle properties are empty. By default, the LongReach app will place the word "Location" into title and the words "Lat,Long values" into subtitle. The location file is a simple JSON object, UTF-8 encoded and a default file contains:

```
{"annotations":[{"title":"Location ","lng":"151.207471","lat":"-33.834345","subtitle":"Lat,Long values"}]}
```

Where the values "lng":"151.207471","lat":"-33.834345" represent a location in North Sydney, Australia.

Programs running on a server can modify and/or generate location files and set the values. Users can transfer these location files to their mobile devices and view the locations on a map. An example of an amended location file is:

```
{"annotations":[{"title":"North Sydney Council","lng":"151.207471","lat":"-33.834345","subtitle":"200 Miller St"}]}
```

Deployment options

LongReach server offers several deployment options:

- Installation on one physical server, accessing files and folders on the same server.
- Installation on multiple physical servers, accessing files and folders on each server.
- Installation on one physical server with connections to remote physical servers, accessing files and folders on the remote servers.
- Installation on one physical server, configured with multiple virtual servers.

Physical servers are any discrete physical, virtual or logically partitioned servers.

In the LongReach app, each server is a discrete entity and the app will have a set of files and folders for each server, regardless of the server deployment methodology.

LongReach with files and folders on one server

The default deployment is LongReach server installed on one physical server (Figure 7, page 25).

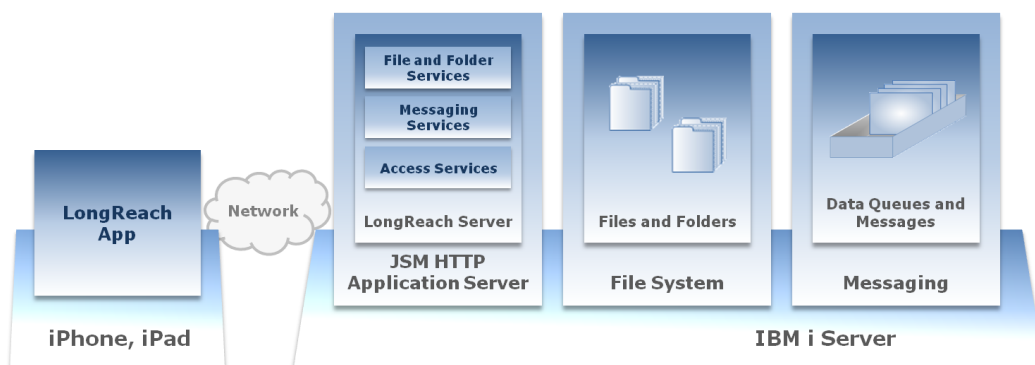


Figure 7: LongReach and File and Folders on one Server

The LongReach server software, files and folders reside on the same server. LongReach app users can configure multiple servers in the app but these configurations all connect to the one physical server where LongReach server resides.

LongReach on multiple servers

Administrators can install LongReach server on multiple physical servers. In this deployment, an instance of the LongReach server software, files and folders resides on each physical server.

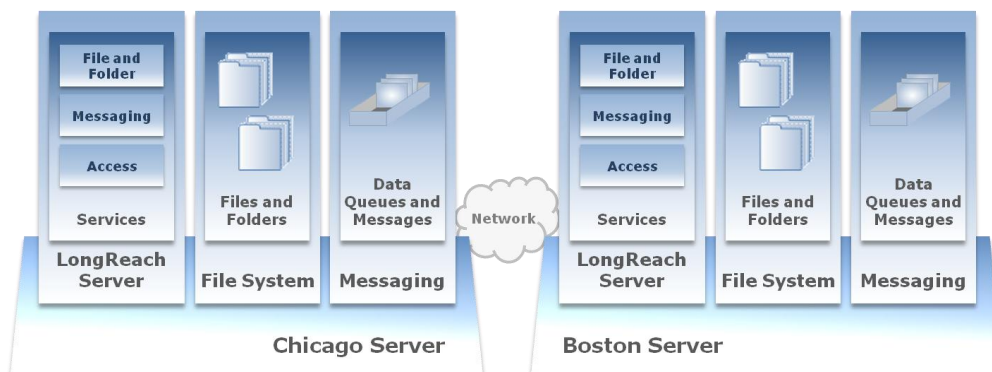


Figure 8: LongReach on Multiple Servers

Figure 8 (page 26) shows LongReach installed on both the Chicago and Boston servers. LongReach app users configure multiple servers in the app and access each server to send and receive files and folders between the app and one of the servers.

LongReach on one server accessing remote servers

LongReach server can access files and folders on remote physical servers without a LongReach installation on the remote servers. Therefore, administrators can install LongReach on one physical server and configure it to access files and folders on one or more remote servers. This deployment provides a way to separate the LongReach server software from the files and folders. It is useful in situations where users require access to files and folders on multiple physical servers but it is impractical to deploy LongReach server on each physical server.

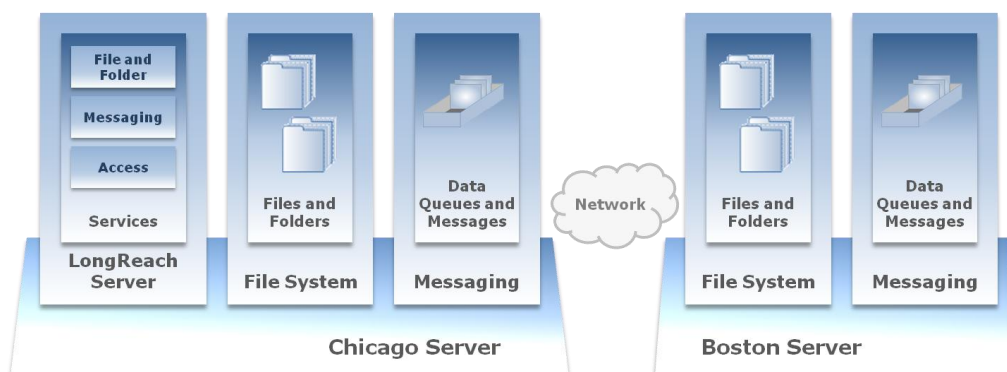


Figure 9: LongReach on one Server with Files and Folders on both Servers

Figure 9 (page 26) illustrates LongReach installed on the Chicago server with a connection to the Boston server. Users can access files and folders on either or both servers. Use this deployment when users require access to files and folders on all servers.

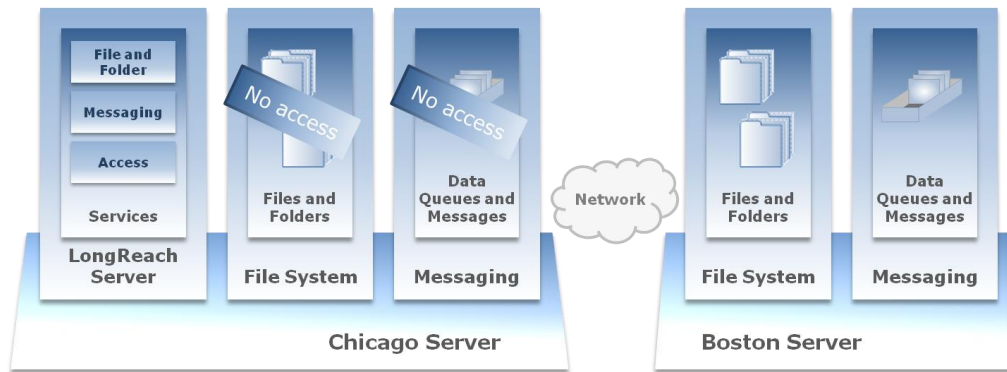


Figure 10: LongReach on one Server with Files and Folders only on another Server

Figure 10 (page 27) illustrates LongReach installed on the Chicago server with a connection to the Boston server. Users can access files and folders only on the Boston server. Use this deployment when you want to separate the LongReach server software from the files and folders or when you can't install LongReach on a server.

Multiple virtual servers

LongReach server can support multiple virtual servers. A virtual server defines the physical server where the LongReach server software resides, a remote physical server or a specific view of any physical server. A view is a collection of users and/or files and folders.

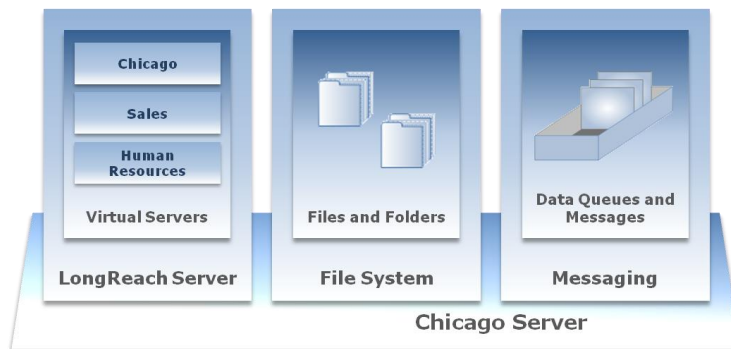


Figure 11: Multiple Virtual Servers in LongReach

Figure 11 (page 27) presents a LongReach server deployment on one physical server (Chicago) and shows the LongReach server, files and folders in the server file system, and messaging resources that manage queues and messages. The configuration defines virtual servers named Chicago, Human Resources and Sales. Different virtual servers define views and access privileges granted to users, files and folders on the Chicago physical server.

Virtual Servers	Users	Files and Folders
Chicago	Available to all users	Users can access all files and folders except human resources and sales files and folders.
Human Resources	Only Human Resources personnel	Only Human Resources personnel can access human resources files and folders.
Sales	Only Sales personnel	Only Sales personnel can access sales files and folders.

Virtual servers allow administrators to simplify assigning access privileges to files and folders for specific users or groups of users. Administrators can use allow and deny directives in the

LongReach configuration to control access privileges granted or denied to users, files and folders. It is possible to configure all allow and deny directives to control users, and file and folder access in one server. However, this configuration will become more complex, and hence more error prone, as the number of users, files and folders increases. Using virtual servers, administrators can grant access to one group of users and deny access to all other users in a few allow and deny directives.

In deployments where LongReach server accesses file and folders on remote servers, administrators can configure virtual servers on the LongReach physical server as well as virtual servers on the remote physical servers.

The following figures illustrate access privileges using the virtual servers Chicago, Human Resources and Sales.

Figure 12 (page 28) shows the access privileges granted to users, files and folders on the Chicago virtual server.

	Files/Folders		
Users	Other	HR	Sales
Users other than HR and Sales	Full access	No access	No access
Human Resources (HR)	Full access	No access	No access
Sales	Full access	No access	No access

Figure 12: Chicago Virtual Server Access Privileges

All users have access to the Chicago virtual server and all users can access files and folders, other than those belonging to human resources and sales.

Figure 13 (page 28) shows the access privileges granted to users, files and folders on the Human Resources virtual server.

HR	Files/Folders		
Users	Other	HR	Sales
Other	No access	No access	No access
Human Resources (HR)	Full access	Full access	No access
Sales	No access	No access	No access

Figure 13: Human Resources Virtual Server Access Privileges

Only human resources personnel can use this virtual server. They can access human resources as well as other files and folders but not sales files and folders.

Figure 14 (page 28) shows the access privileges granted to users, files and folders on the Sales virtual server.

Sales	Files/Folders		
Users other than HR and Sales	Other	HR	Sales
Other	No access	No access	No access
Human Resources (HR)	No access	No access	No access
Sales	Full access	No access	Full access

Figure 14: Sales Virtual Server Access Privileges

Only sales personnel can use this virtual server. They can access sales as well as other files and folders but not human resources files and folders.

Configuring remote and virtual servers

Administrators can configure remote and/or virtual servers in the httpd.xml configuration file by creating host directives.

Host directives

The LongReach default configuration includes the settings for the server that will host the LongReach server software. This physical server is the local host and other physical servers are remote hosts.

The host directive in the LongReach service section of the httpd.xml configuration file is where you configure remote and virtual servers. The server that hosts LongReach server can be configured as LOCALHOST; this is the default configuration. Insert a host directive and its parameters for each additional server.

Remote activation key

LongReach server requires a remote activation key to enable access to remote physical servers. Remote activation is a parameter of the LongReach service. One remote activation key enables multiple remote servers, including virtual servers on remote servers.

No remote activation key is necessary when using LongReach on only one physical server.

No remote activation key is necessary when using LongReach on multiple physical servers.

No remote activation key is necessary when using LongReach virtual servers on physical servers where LongReach resides.

Prerequisite software

All servers require installation of the IBM Toolkit for Java.

The LongReach server software does not reside on remote servers in this deployment.

Accessing files and folders on remote servers

Each server has its own LongReach file and folder structure, including longreachdata/shared and longreachdata/user/[user-identification]. LongReach creates the folder structure the first time any user sends files to the server. Administrators do not need to create the longreachdata folder on remote servers.

On a mobile device each physical server has its own hierarchical file structure.

To access a physical server users configure the profile in the server configuration of the LongReach app using a fully qualified user-identification (Figure 15, page 29).

Physical Server	Fully Qualified User Identification	Example: user JOHND
Boston	Boston/[user-identification]	Boston/JOHND
Chicago	Chicago/[user-identification]	Chicago/JOHND

Figure 15: How Users Access a Physical Server

This example assumes that JOHND has the same user-identification on both physical servers.

Section 3 - Administration

This section provides information about LongReach that will help administrators to plan and manage LongReach effectively.

Administrator responsibilities

Administrator responsibilities are:

- Installing LongReach on the server
- Configuring LongReach server components
- Providing communications configuration information to LongReach app users
- Configuring user access permissions to a LongReach server
- Configuring mobile device access permissions to a LongReach server
- Configuring file and folder access permissions
- Managing user requests and questions
- Ensuring regular backups occur
- Troubleshooting

Installation and server configuration are once only or occasional tasks. Housekeeping is an ongoing task. Troubleshooting is an occasional task.

Managing users and devices is an ongoing task as administrators are responsible for servicing requests for user profiles, user access, and mobile device access.

Managing file and folder access permissions is an ongoing task.

Users require communications information to use LongReach server and administrators are responsible for providing this information. Users will not be able to connect to a LongReach server without the communications configuration items.

Users are responsible for managing the LongReach app on their mobile devices.

Administrators do not need to manage the LongReach app unless they provide mobile device support for company employees and contractors.

Decisions administrators need to make

This section describes issues you need to think through and decisions you must take before installing and configuring LongReach server components. Refer to the LongReach installation instructions for more detailed information about installing a LongReach server.

Port numbers

Select the port or ports the JSM HTTP Server will use for LongReach.

The default ports are 6563 when using HTTP or 6564 when using HTTPS (TLS/SSL).

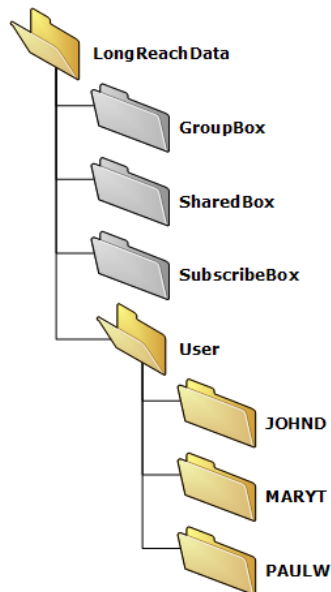
Device count

Choose a number as the value for the maximum device count.

Folder structure and naming conventions

LongReach installs a folder structure that will contain a folder for each user and special purpose folders. A user folder will contain files and sub-folders belonging to the user who

owns the folder. LongReach creates the user folder the first time a user connects to the LongReach server.



The default top folder name is: longreachdata

The folder longreachdata/user is the top folder for user files and folders. Administrators may use a different folder name for the top folder by amending the configuration file before allowing users to connect.

User folder name format is:

longreachdata/user/username

Folders represented by username are the user folders. In this example the users are JOHND, MARYT and PAULW.

For example, longreachdata/user/JOHND is the name of JOHND's user folder. The folder name derives from the user profile defined in the communications settings on the mobile device.

An advantage of grouping user files and folders under a specific top folder is simplicity of backup at either the top folder level or performing backups for individual users.

Warning

Do not use the root folder in the IFS as the top folder for LongReach.

File transfer notifications

LongReach can send a notification to a data queue on an IBM i server each time a file transfer occurs. Administrators need to design a naming convention and plan for data queue management.

Prerequisites

The LongReach Installation Instructions define the prerequisites for installing and operating a LongReach server.

Important reminder

The LongReach server will not operate successfully without the prerequisite software.

Install LongReach server components

Installation and configuration process

Refer to the LongReach installation instructions for detailed information about the installation procedure. The steps in the installation and configuration process are:

1. Read the documentation.
2. Install the prerequisite software on the server.
3. Install LongReach on the server.
4. Apply licences (not necessary for the trail period).

- | | |
|----|--------------------------------|
| 5. | Configure the JSM HTTP Server. |
| 6. | Configure LongReach services. |

You will reduce your workload if you are familiar with the options for configuration before you start.

New installations

During installation, LongReach copies the template configuration file named `httpd-template.xml` to an instance configuration named `httpd.xml`. This file contains default configuration settings.

You can amend the configuration if necessary; before amending items in the configuration file, make a backup of the `httpd.xml` file.

Upgrades

During upgrades LongReach will add components, amend installed components and remove deprecated components. Upgrades will replace the existing copy of the configuration file named `httpd-template.xml`. Therefore, it is essential that you copy and backup configuration files before installing upgrades.

Upgrades will not replace the existing `httpd.xml` configuration file. However, upgrades may require changes to individual configuration items in the file. You should ensure that your copy of the `httpd.xml` file includes changes made by the upgrade by comparing the contents of the new `httpd-template.xml` file with your existing `httpd.xml` file.

Configuration files and their locations

Configuration files reside in folders (or directories) in the file system on the server. On IBM i servers the file system is the IFS. Table 5 (page 32) provides the locations of the configuration files.

Table 5: Configuration File Locations

Configuration Files	Locations
<code>httpd.xml</code>	<code>longreach/jsm/instance/system/</code>
<code>httpd-template.xml</code>	<code>longreach/jsm/instance/system/</code>
<code>manager.properties</code>	<code>longreach/jsm/instance/system/</code>
<code>device-longreach.txt</code>	<code>longreach/jsm/instance/system/</code>

The root folder (`longreach`) in the location is the folder name chosen on installation.

What to configure for LongReach

The configuration section of this guide describes each configuration item in detail for both the JSM HTTP Server and LongReach services. Any installation of LongReach may require changes to the default configuration. This section identifies which configuration items to configure.

JSM HTTP Server

Table 6 (page 33) identifies sections in the JSM HTTP Server configuration that may change if the default configuration is inappropriate for a LongReach installation. Change for most

items is optional. The port numbers are mandatory if the default port numbers are unavailable.

Table 6: What to Configure for the Server Instance

To configure	Apply these settings	Change default
Access logging	Set accesslog enabled to true. Define the access log file name.	Optional
Active	Active must be true	Do not change
Backlog	Choose the depth of the TCP/IP queue.	Optional
Buffer receive	Set the size (bytes) of the receive buffer	Optional
Buffer send	Set the size (bytes) of the send buffer	Optional
Error logging	Set errorlog enabled to true. Define the error log file name.	Optional
Index	Define the index document	Optional
Instance name	Choose a name	Optional
Interface	Use *ALL for all addresses or choose the interface address. Default is all interfaces.	Optional
No delay	Set to True to enable TCP/IP no delay. False uses the operating system settings.	Optional
No TLS/SSL	Set listen secure to false. When listen secure is false, the JSM HTTP Server ignores store and password values.	Optional
Port	Insert the port number you want to use.	Mandatory The server expects either Port and/or SSLport.
Root	Define the root directory (folder)	Optional
Secure connection with TLS/SSL	Set listen secure to true. Define the name of the store. Insert the password.	Optional
Service folder	Insert allow and deny directives for folders.	Optional Mandatory if you need to control access to specific files and folders.
Service user	Insert allow and deny directives for users.	Optional

To configure	Apply these settings	Change default
SSLport	Insert the port number you want to use.	Mandatory when secure connection is true. LongReach expects either Port and/or SSLport.
Timeout	Set the connection timeout in seconds.	Optional

Whether you need to change the optional items depends on the environment of your LongReach installation and security and/or performance requirements.

LongReach services

Once the LongReach server is installed, configured and operational administrators need to consider what changes are necessary to the default access permissions for users, devices, files and folders. It is unlikely that the default access permissions will be adequate for all circumstances.

User access permissions

The tasks to implement user access permissions are:

1.	Construct a list of the people who will be allowed to use the LongReach server.
2.	Identify people who will not be allowed to use the LongReach server.
3.	Insert service.user.deny and server.user.allow directives into the LongReach configuration file that define the access permissions.

Device access permissions

The tasks to implement device access permissions are:

1.	Configure the maximum number of devices in the device-longreach.txt file.
2.	LongReach will automatically register devices as they connect to the LongReach server, until the device count equals the maximum devices.

If you do not wish to limit the number of devices that can connect to a LongReach server, choose a large number as the value of the maximum number of devices.

Folder access permissions

Administrators do not need to include folder get, put and delete deny and allow directives for user folders.

The tasks to implement file and folder user access permissions are:

1.	Construct a list of folders that require controlled access.
2.	Identify access permission to apply to each folder.
3.	Insert service.folder.get, put and delete deny and server.folder.get, put and delete allow directives into the LongReach configuration file to implement the access permissions.

Creating a set of complex folder access permissions will increase the maintenance workload for administrators.

Using HTTP or HTTPS with LongReach

Communications between a LongReach app and a LongReach server using HTTP offers a strong level of protection. LongReach HTTP connections use an RSA 1024-bit asymmetric key and RC4 128-bit symmetric key mechanism similar to SSL to encrypt the data transferred between a LongReach app and a LongReach server.

LongReach HTTPS/SSL connections use the LongReach data encryption mechanism as well as the additional protection offered by SSL. You can also use a HTTPS/SSL connection from the LongReach app to a proxy server (IIS, Apache) and then change to an internal HTTP connection from the proxy server to the LongReach server.

To configure LongReach to use HTTPS/SSL you need to prepare a private/public key and certificates, and change settings in the LongReach httpd.xml configuration file.

Public/private keys and certificates

The X.509 certificate must have a domain name in its alternate subject name extension (AltSubjectName). The domain name is the DNS value used for the URL to connect to the LongReach server via HTTPS. For example, using the LongReach URL <https://mobile.mycompany.com/service/longreach.jsp> the value in the AltSubjectName is:

mobile.mycompany.com (explicit name)

*.mycompany.com (wildcard name)

You can use an existing private/public key signed by a well known certifying authority (VeriSign); or create a self-signed private/public key.

The LongReach SSL server requires a private key and certificates chain in a JKS keystore file.

To use a key with LongReach, export the key to a JKS keystore with a single entry private key and certificates chain. The LongReach SSL server uses only the first entry in the keystore, and the alias name is mandatory and can be any string value.

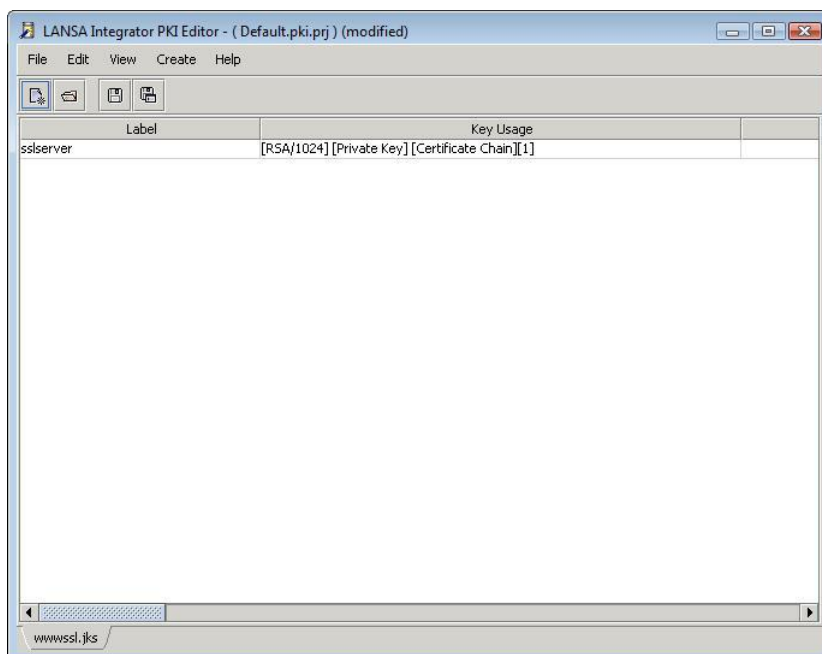


Figure 16: LANSA Integrator PKI Editor

Figure 16 (page 35) is an example of the content of a single entry keystore. The column titled Label is the alias, and in this example the alias is sslserver.

The LANSA Integrator PKI Editor is not supplied with LongReach.

Configure LongReach for SSL

Edit the listen element of the instance directive in the LongReach configuration file.

```
<listen timeout="5" nodelay="false" bufferreceive="-1" buffersend="-1"
password="password" store="pki/wwwssl.jks" secure="true" backlog="256"
interface="*ALL" sslport="6564" port="6563"/>
```

The properties to change are:

Properties	Value	Explanation
secure	secure="true"	Set the value of this property to true (the default value is false).
sslport	sslport="6564"	Set the value of this property to the port you wish to use for HTTPS/SSL. The default value is 6564.
store	store="pki/wwwssl.jks"	This is the default keystore. Change this property if you used a different name for the keystore file.

The SSL private key and certificates chain come from the file specified in the store property.

Manage log and error files

This section explains administrator tasks for managing access log files and error log files. These files accumulate with use and administrators should archive and/or clear the files periodically.

File names and locations

In a typical installation log files reside in a logging folder.

Location	longreach/jsm/instance/www/instance/logs/
Log File Names	error.log access.log longreach.log

Examining the content of these files will reveal information about activity and errors.

Sample longreach.log	[4][10.2.1.42] [2011-05-18 14:33:03 +1000] [LOCALHOST] [MyUser] [getfile] [/devjsm/instance] [longreach-get.pdf] [6][10.2.1.42] [2011-05-18 14:33:16 +1000] [LOCALHOST] [MyUser] [getlist] [/devjsm/instance] [8][10.2.1.42] [2011-05-18 14:33:28 +1000] [LOCALHOST] [MyUser] [putfile] [/devjsm/instance] [longreach-put.pdf]
-----------------------------	--

Enable and disable logging

The errorlog and accesslog directives in the configuration file control logging activity.

To enable logging for both error and access logs, set the value of the enabled parameter as true:

```
<errorlog enabled="true" file="www/instance/logs/error.log"/>
```

```
<accesslog enabled="true" file="www/instance/logs/access.log"/>
```

To disable logging for both error and access logs, set the value of the enabled parameter as false:

```
<errorlog enabled="false" file="www/instance/logs/error.log"/>
```

```
<accesslog enabled="false" file="www/instance/logs/access.log"/>
```

Error and access logs operate independently. For example you can enable error logging and disable access logging.

Archive log files

Archive log files by copying them to another location.

LongReach provides no archiving tools.

Delete log files

Do not delete the active error.log and access.log files.

LongReach will archive the active log files periodically. Before deleting the archived log files, ensure you either have a backup or archive copy of the files.

Collect data to assist in finding error causes

This section explains the administrator tasks for collecting data to assist with tracking errors. Administrators need to enable and disable tracing services and clear trace files when they are no longer needed.

Tracing allows you to collect information about the activity of LongReach. When enabled, tracing directs STDOUT and STDERR to the trace files STDOUT.TXT and STDERR.TXT.

The Java Virtual Machine and instance information is logged to a MANAGER.TXT file.

Trace files: names and locations

In a typical installation trace files reside in two folders.

The first location provides high level trace information.

Location	longreach/jsm/instance/trace/[job number]/
Log File Names	CLASSPATH.TXT MANAGER.TXT STDERR.TXT STDOUT.TXT
Sample	longreach/jsm/instance/trace/189002/STDOUT.TXT

The [job number] is the job number of the active job when tracing occurred.

The second location provides detailed trace information.

Location	longreach/jsm/instance/trace/[job number]/[date]/
Log File Directory Names	HTTP00000000 HTTP00000001 HTTP00000002 HTTP00000003

Sample

longreach/jsm/instance/trace/309001/2009-11-24/HTTP00000004

Enable and disable tracing

The trace and clienttrace directives control tracing activity. These directives are in the virtual host sections of the configuration file and are associated with each service. Administrators can enable tracing from the server (server-initiated traces) or clients can request tracing (client-initiated traces). Tracing directives operate independently, for example, you can enable tracing for one service or all services and/or enable or disable client-initiated tracing.

Server-initiated traces

Server-initiated tracing generates tracing information for all clients of a service.

To enable tracing for a service, set the value of the trace parameter as true:

```
trace="true" clienttrace="false"
```

Client tracing is disabled and unnecessary in this example because tracing is enabled from the server.

To disable tracing for a service, set the value of the trace parameter as false:

```
trace="false" clienttrace="false"
```

This example shows how to disable tracing completely. The server will not perform tracing and clients cannot request tracing.

Client-initiated traces

The clienttrace directive allows clients to initiate tracing. Clients request tracing by appending ?trace=true to the URL. The clienttrace directive must also be enabled.

Client-initiated tracing allows administrators to collect trace information for specific clients for one session or for many sessions. Tracing for a client will occur whenever the client includes ?trace=true and client-initiated tracing is enabled.

To enable client-initiated tracing for a service, set the value of clienttrace as true:

```
trace="false" clienttrace="true"
```

In this example server tracing is disabled and client tracing is enabled.

LongReach will generate traces requested by clients only when both the value of the clienttrace directive is true and the URL includes ?trace=true. When clienttrace is false clients cannot generate tracing even when the URL includes ?trace=true.

Clear trace files

Use the command CLRJSM command to clear trace files and subdirectories. The command has the following parameters:

INSTANCE	The instance defaults to a value of *DEFAULT. This is the recommended value.
TRACEDIR	The option defaults to a value of *YES. Valid values are *YES, *NO. The value *YES removes files and subdirectories in the trace directory.

TEMPDIR	<p>The option defaults to a value of *YES.</p> <p>Valid values are *YES, *NO.</p> <p>The value *YES removes all files and sub-directories in the TEMP directory.</p> <p>Do not clear the TEMP directory while JSM services are running, as this action will delete temporary files used by the services. Always use *NO when clearing trace files from a running JSM instance.</p>
KEEP	<p>Valid values are in the range 0 to 99 days.</p> <p>Zero means do not keep the files. Running the command will delete ALL files and sub-directories.</p> <p>Use 1 to keep only today's files.</p> <p>Use 2 to keep today's and yesterday's files.</p>

Archive trace files

Archive trace files by copying them to another location, before clearing the files.

LongReach provides no archiving tools.

Troubleshooting

The troubleshooting section suggests the parts of LongReach to check when it does not operate as expected.

Error messages

Table 7 (page 39) describes error messages, their cause and how to fix the problem.

Table 7: Error Messages

Messages	Explanation and Remedy
Mobile app version is not supported, upgrade mobile app.	<p>The LongReach app version is not supported by the version of the LongReach server that the app is trying to use.</p> <p>The version of the LongReach app is superseded and incompatible with the version of the LongReach server.</p> <p>The remedy for this problem is to upgrade the LongReach app.</p>
Mobile app version is not supported by the server, upgrade server version.	<p>The LongReach app is attempting to connect to an older version of the LongReach server.</p> <p>The version of the LongReach server is superseded and incompatible with the version of the LongReach app.</p> <p>The remedy for this problem is to upgrade the LongReach server.</p>

Installation and configuration

JSM HTTP Server does not start

- | | |
|----|--|
| 1. | Check the manager.properties httpd directive:
httpd=system/httpd.xml
Ensure that the name specified for the httpd configuration file is the name of the actual httpd configuration file. |
| 2. | Check the JSM trace file MANAGER.TXT for JSM HTTP Server start up messages. |
| 3. | Review the configuration directives in the instance tag. |
| 4. | Confirm that the JSM HTTP Server is started by using a browser to access the home index.html file. |

Ensure that the configuration is complete before attempting to start the JSM HTTP Server.

IBM Toolbox for Java not installed

- | | |
|----|--|
| 1. | Ensure that the IBM Toolbox for Java is installed.
Verify that the file, "jt400.jar", resides in the JSM jar directory. |
|----|--|

Installing the IBM Toolbox for Java is an essential prerequisite for operating LongReach.

File ownership and permissions for files in the IFS

- | | |
|----|---|
| 1. | The JSM HTTP Server by default runs as the JSM job description user profile.
Any files and directories created should be owned by that user profile. |
| 2. | Verify the user profile, file and directory ownership. |
| 3. | Change ownerships if necessary. |

Get support from the LongReach Forums

The LongReach Forums (<http://longreach.lansa.com.au/>) provide an opportunity to participate in the LongReach community by asking questions, providing answers and making suggestions.

LongReach Technical Support Q&A Forum	Use this forum to post technical support questions and/or answer questions posted by other LongReach users.
LongReach Tips and Techniques Forum	Use this forum to post tips and techniques on how to use LongReach.
LongReach Suggestion Box	Use this Forum to suggest ideas for improvements and new features.
LongReach Product Announcements and Release Information	LANSA uses this space to inform LongReach users about new versions, features and product releases.
Sand Box for Testing LongReach	Test the LongReach server before you download and install.

The Sand Box is useful for testing the LongReach server when you don't have access to an IBM i server. Send an email to longreach.sandbox@lansa.com.au and we will send you the information you need to access LANSA's LongReach Sandbox.

Section 4 - Configuration

This section explains the tools and configuration options available to administrators who will manage LongReach.

The configuration process

The steps in the configuration process are:

1.	Copy and rename the configuration file: <code>httpd-template.xml</code> . The recommended name for the copied file is: <code>httpd.xml</code> You can use any valid name but you should choose a name that indicates the file content.
2.	Change the <code>httpd</code> property in the <code>manager.properties</code> file to the new name allocated when you copied the original <code>httpd</code> configuration file. For example, if you used the recommended name for the copied <code>httpd</code> configuration file, change the <code>httpd</code> property in <code>manager.properties</code> to: <code>httpd=system/httpd.xml</code> If not, use the name allocated to the copied file.
3.	Configure the JSM HTTP Server instance.
4.	Configure the LongReach services you are licensed to use.

The configuration process assumes you have completed the installation successfully.

Important reminders

Always make a copy of the `httpd` configuration file before you edit its contents.

Keep a separate copy of the `httpd` configuration file that is active in your production environment.

Use a text editor or XML editor when changing configuration items and parameters in the `httpd` configuration file. Exercise care when editing and make sure you do not change tag names.

About the configuration files

The configuration details reside in these files: `manager.properties` and `httpd.xml`. This section describes the content of the configuration files. Later sections describe how to change individual configuration items.

`manager.properties`

The properties file, `manager.properties`, contains information about the installed instance of LongReach. Table 8 (page 42) explains the properties `manager.properties`.

Table 8: Explanation of `manager.properties`

Properties	Definitions
<code>httpd=system/httpd.xml</code>	Name of the <code>httpd</code> configuration file

After updating the httpd configuration file property in manager.properties, the configuration file will look like the following example (Table 9, page 43).

Table 9: Example of an Updated manager.properties File

Properties
httpd=system/httpd-template.xml httpd=system/httpd.xml # tcp.port=6560 console.tcp.port=6561 studio.client.address=*none console.client.address=*none

The line beginning with a hash (#) is a comment line.
The file manager.properties resides in the folder longreach/jsm/instance/system/.

httpd configuration file

The configuration file named httpd.xml contains the configuration items and parameters for LongReach. The file exists in two forms. The files httpd-template.xml contains default setting and is the template for the operational file named httpd.xml.

Table 10: Structure of the httpd Configuration File

Structure of httpd Configuration File
<?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5" /> + <access> + <mimetype> + <virtual host="*" active="true"> </instance> </configuration>

Table 10 (page 43) shows the structure of the httpd configuration file.
The configuration contains an instance of the JSM HTTP Server. The instance has a set of configuration parameters (instance, errorlog, accesslog and listen tags). The instance includes configuration items for access, MIME types (or Internet Media Types) and virtual hosts.

Table 11 (page 44) provides a list of recommendations for changing configuration items and parameters in the httpd configuration file.

Table 11: Mandatory and Optional Changes to Configuration Items and Parameters

Configuration Items	Change Mandatory/Optional
JSMHTTPServiceFile (match URI, class)	Do not change these items.
MIME types (server instance)	Optional, but not recommended.
MIME types (virtual host)	Optional, but not recommended.
Ports	You must choose port numbers for HTTP and/or HTTPS, or accept the default ports.
Realm (protect tag, virtual host)	Do not change any items in the protect tag.
Virtual host name and active	Do not change these items unless you choose multiple instances and/or multiple virtual hosts.

The default values of many configuration items and parameters in the httpd configuration will be appropriate for operating LongReach. Some values you must change and some are optional.

Important reminders	<p>Copy the original httpd configuration file.</p> <p>Do not use the original httpd-template.xml configuration file for your configuration settings. Version upgrades and fixes may alter the parameters in this configuration file and will over-write your configuration settings when installed.</p>
----------------------------	---

Ports

The httpd configuration file contains items to define the ports LongReach will use. Table 12 (page 44) shows the default ports.

Table 12: Default Ports

Port Numbers	Definitions
6560	The JSM Manager TCP port is used internally and it does not accept TCP/IP connections.
6561	The JSM Console TCP port is used internally and it does not accept TCP/IP connections.
6563	JSM HTTP Server (default port for LongReach without TLS/SSL)
6564	JSM HTTPS Server (default port for LongReach with TLS/SSL)

To change ports follow these steps:

1.	Edit httpd.xml (or the equivalent file if you use a different name).
2.	Under the listen tag of the instance: Change the port parameter. Change the sslport parameter.
3.	Save httpd.xml

The configuration files shipped with LongReach are configured to use the default ports. You do not need to change the port configurations if the default ports are suitable for your installation.

JSM HTTP Server

This part of the administrator guide explains how to configure the sections of the httpd configuration file that apply to the JSM HTTP Server and LongReach. There are separate sections that explain how to configure individual services that the JSM HTTP Server supports.

Server instance configuration

Table 13 (page 45) shows the set of configuration items and parameters that apply to the JSM HTTP Server instance.

Table 13: Configure a Server Instance

Configure Server Instance Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> + <access> + <mimetypes> + <virtual host="*" active="true"> </instance> </configuration></pre>

In this example, the instance of LongReach has error logging and access logging enabled (the value of the parameter is "true"). The server will place errors into error.log and access events into access.log. The instance will listen on port 6563.

Controlling access to the server instance

The access directive (at the instance level) specifies rules for accessing the instance of the JSM HTTP Server by allowing and/or denying addresses. You use combinations of allow and

deny directives to control access to the server instance. Addresses can be specific (10.2.45.1), masks (10.2) or generic (indicated by the asterisk (*)).

Table 14 (page 46) shows an example of allowing any address `<allow address="*" />`.

Table 14: Control Access to the Server Instance

Server Instance Access (Allow and Deny) Configuration Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> <access> <allow address="*" /> </access> +<mimetype> +<virtual host="*" active="true"> </instance> </configuration></pre>

Any number and of allow and deny directives are permitted.

The JSM HTTP Server ignores allow and deny directives inside comments.

Do not use directives for user agents or content lengths in this section.

Table 15 (page 46) provides examples of allow and deny directives.

Table 15: What to Configure for Server Instance Access

To configure	Apply these settings	Change defaults
Allow any address	Use <code>allow address=*</code> <code><allow address="*" /></code>	Optional
Access for specific addresses	Add new a directive for each allowed address. <code><allow address="10.2.1.45"/></code>	Optional
Deny any address	Use <code>deny address=*</code> <code><deny address="*" /></code>	Optional
Deny access for specific addresses	Add new a directive for each denied address. <code><deny address="10.2.1.45"/></code>	Optional

Changing the default settings for server instance access is optional unless access requires specific address exclusions (deny).

MIME types for the server instance

Table 16 (page 47) shows the configuration for MIME types. This part of the httpd configuration file defines MIME types applicable to the whole instance.

The MIME type directives allow the JSM HTTP Server to correctly understand the nature of files. Administrators do not need to change these directives.

Table 16: MIME Types for the Server Instance

Server Instance MIME Types Configuration Example
<pre> <?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> + <access> <mimetype> <map extension="png" type="image/png"/> <map extension="gif" type="image/gif"/> <map extension="jpg" type="image/jpeg"/> <map extension="jpeg" type="image/jpeg"/> <map extension="tiff" type="image/tiff"/> <map extension="ico" type="image/x-icon"/> <map extension="pdf" type="application/pdf"/> <map extension="css" type="text/css; charset=utf-8"/> <map extension="xsl" type="text/xls; charset=utf-8"/> <map extension="xml" type="text/xml; charset=utf-8"/> <map extension="htm" type="text/html; charset=utf-8"/> <map extension="html" type="text/html; charset=utf-8"/> <map extension="js" type="application/x-javascript; charset=utf-8"/> </mimetype> + <virtual host="*" active="true"> </instance> </configuration> </pre>

MIME type directives in the virtual host section of the configuration file override MIME types specified in this section.

To determine allowed MIME types, the JSM HTTP Server looks at MIME types in the virtual host section of the httpd configuration file and then at MIME types in the server instance. Place MIME types in the server instance that will apply all virtual hosts. Place MIME types that are unique to a virtual host in the MIME types section of the virtual host.

Table 17: What to Configure for the Server Instance MIME Types

To configure	Apply these settings	Change defaults
MIME types	Use the default list	Not recommended
Add MIME types	Add new a directive for each MIME type.	Optional

Virtual host configuration

The JSM HTTP Server is capable of managing multiple virtual hosts. Table 18 (page 48) shows the virtual host configuration for LongReach.

Table 18: Virtual Host Configuration

Virtual Host Configuration Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> +<access> +<mimetype> <virtual host="*" active="true"> +<access> +<protect> +<script> +<mimetype> </virtual> </instance> </configuration></pre>

The host is the name of the virtual host to match with the HTTP host property. In this case the asterisk ("*") indicates acceptance of requests from any host.

Table 19: What to Configure for the Virtual Host

To configure	Apply these settings	Change defaults
Virtual host	Use the default value asterisk (*)	Do not change
Active	Use the value true	Do not change

The active parameter has two values "true" and "false". The value of the active parameter must be "true" for LongReach to operate.

Access, protect, script and MIME type are sub sections of the virtual section in the httpd configuration file.

Virtual host access

The access directives in the virtual host section in the httpd configuration file control access to services provided by the JSM HTTP Server. These directives override the access directives defined for the server instance. Using the virtual host access configuration you can:

- Allow and/or deny addresses
- Allow and/or deny user agents
- Allow and/or deny content lengths

Table 20 (page 49) shows an example of configuring virtual host access directives.

Table 20: Virtual Host Access Directives

Virtual Host Access Directives Configuration Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> +<access> +<mimetype> <virtual host="*" active="true"> <access> <allow address="*" /> <deny useragent="android" /> <deny useragent="ipad" /> <deny useragent="iphone" /> <allow useragent="*" /> <allow useragent="?" /> <deny contentlength="4096" /> <!-- deny content GT value --> </access> +<protect> +<script> +<mimetype> </virtual> </instance> </configuration></pre>

In this example, requests from any address are acceptable, deny access is explicit for several user agents, other user agents are acceptable and content length greater than 4096 is unacceptable.

Table 21: What to Configure for Virtual Host Access

To configure	Apply these settings	Change defaults
Allow any address	Use allow address=*	Optional
Access for specific addresses	Add a directive for each address.	Optional
Allow any user agent	Use allow useragent=*	Optional
Allow specific user agents	Add a directive for each user agent	Optional
Allow content lengths	Use allow contentlength= "value"	Optional
Deny any address	Use deny address=*	Optional
Deny access for specific addresses	Add a directive for each address.	Optional
Deny any user agent	Use deny useragent=*	Optional
Deny specific user agents	Add a directive for each user agent	Optional
Deny content lengths	Use deny contentlength= "value"	Optional

Access directives in the virtual host section of the configuration file override access directives specified in the access section of the server instance.

Virtual host protect

Warning

You do not need to configure this section of the httpd configuration file to use LongReach, unless you wish to change the default settings.

The protection section of the httpd configuration file maps authentication methods to parts of the Web site or application. The realm describes the authentication method and the match URI associates the realm with the protected part of the Web site or application.

Table 22 (page 51) shows an example of the protect configuration.

Table 22: Virtual Host Protect Configuration

Virtual Host Protect Configuration Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> + <access> + <mimetype> <virtual host="*" active="true"> + <access> <protect> <realm name="Area 51"> <user name="user" access="bb644a9819425bfd8586b408896a1031"/> </realm> <match uri="/restricted" realm="Area 51" authentication="basic,digest"/> </protect> + <script> + <mimetype> </virtual> </instance> </configuration></pre>

In this example the realm is "Area 51", the user name is "user", and access is a hash of the user, password and realm information. The configuration uses both basic and digest authentication methods. The match uri = "/restricted" associates the realm with URIs including the match URI.

Table 23: Configuring Virtual Host Protect

To configure	Apply these settings	Change defaults
Realm	Define the realm name.	Do not change the default values.
Realm / user	Add one or more user names with their access hash into the realm. You need to generate an access hash for each user in each realm and include them in the configuration. This applies even when the same user is in different realms. The tool to generate the access hash is not supplied with this version of LongReach.	Do not change the default values.
Match URIs	Add one or more full or partial URIs with their associated realms.	Do not change the default values.

You do not need to configure this section of the httpd configuration file to use LongReach.

Virtual host script

The script section of the httpd configuration file contains configuration items associated with the available services. Table 24 (page 52) shows an example of a script configuration.

Table 24: Virtual Host Script Configuration

Virtual Host Script Configuration Example
<pre> <?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> + <access> + <mimetype> <virtual host="*" active="true"> + <access> + <protect> <script> <match uri="/service/longreach.jsp" class="com.lansa.mobile.service.HTTPServiceLongReach" trace="true" clienttrace="false"> <parameter name="service.default.host" value="LOCAL"/> <parameter name="service.remote.activation" value=" </pre>

Virtual Host Script Configuration Example

```

7F04E051D3D9F04458D348832BE7CEFE15572A0F"/>
<host name="LOCAL" system="LOCALHOST"/>
  <parameter name="service.access.log"
    value="www/instance/logs/longreach.log"/>
  <parameter name="service.user.deny" value="*USER"/>
</host>
<host name="CHICAGO" system="LOCALHOST"/>
  <parameter name="service.access.log"
    value="www/instance/logs/longreach.log"/>
  <parameter name="service.user.allow" value="*USER"/>
  <parameter name="service.folder.base"
    value="/longreachdata/user/{NAME}"/>
  <parameter name="service.folder.shared"
    value="/longreachdata/shared"/>
  <parameter name="service.folder.get.allow" value="{BASE}"/>
  <parameter name="service.folder.get.allow" value="{BASE}/*"/>
  <parameter name="service.folder.put.deny"
    value="{BASE}/manuals"/>
  <parameter name="service.folder.put.deny"
    value="{BASE}/manuals/*">
  <parameter name="service.folder.put.allow" value="{BASE}"/>
  <parameter name="service.folder.put.allow" value="{BASE}/*"/>
  <parameter name="service.folder.delete.deny"
    value="{BASE}/manuals"/>
  <parameter name="service.folder.delete.deny"
    value="{BASE}/manuals/*">
  <parameter name="service.folder.delete.allow" value="{BASE}"/>
  <parameter name="service.folder.delete.allow" value="{BASE}/*"/>
  <notify data="{FOLDER}" key="{NAME}" padkey="true" ccsid="0"
    queue="/QSYS.LIB/LONGREACH.LIB/NOTIFY2.DTAQ"/>
  <!--
  <notify data="{NAME},{FOLDER}" ccsid="0"
    queue="/QSYS.LIB/LONGREACH.LIB/NOTIFY1.DTAQ"/>
  -->
</host>
</match>
</script>

+ <mimetype>
</virtual>
</instance>
</configuration>

```

The configuration file shipped with LongReach may include settings belonging to other services. These services are not licensed for use with LongReach.

Virtual host MIME types

Table 25 (page 54) shows the configuration for MIME types applicable to the virtual host.

Table 25: Virtual Host MIME Type Configuration

Virtual Host MIME Type Configuration Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> + <access> + <mimetype> <virtual host="*" active="true"> + <access> + <protect> + <script> <mimetype> <map extension="pdf" type="application/pdf"/> </mimetype> </virtual> </instance> </configuration></pre>

The MIME types defined in this section of the httpd configuration file override MIME types defined for the server instance.

Table 26: What to Configure for Virtual Host MIME Types

To configure	Apply these settings	Change defaults
MIME types	Use the default list	Do not change
Add MIME types	Add new a directive for each MIME type.	Optional

Access log

Parameters explained in this section are:	service.access.log
---	--------------------

The parameter service.access.log defines the file LongReach will use when logging server access events.

<parameter name="service.access.log" value="www/instance/logs/longreach.log"/>

In this example the access log file is longreach.log and it resides in the logs folder.

User access permissions

Parameters explained in this section are:

- service.user.allow
- service.user.deny

The user.allow and user.deny parameters define user identifications (or profiles) that are either permitted to use LongReach or excluded from LongReach services. The service.user.allow parameter defines acceptable users and the service.user.deny parameter defines users not permitted to use LongReach services. Table 27 (page 55) explains the syntax of these parameters. You may include multiple instances of the service.user.allow and service.user.deny parameters.

Table 27: User Allow and Deny Parameter Syntax

Parameter	Actions
<parameter name="service.user.allow" value="UserId"/>	The parameter value defines allowed (or acceptable) user identifications.
<parameter name="service.user.allow" value="UserId,UserId,UserId,UserId"/>	The parameter value is a list of user identifications separated by commas. The service.user.allow parameter allows (or accepts) user identifications in the list.
<parameter name="service.user.deny" value="UserId"/>	The parameter value denies (excludes) access to one user.
<parameter name="service.user.deny" value="UserId,UserId,UserId,UserId"/>	The parameter value is a list of user identifications separated by commas. The service.user.deny parameter denies (excludes) access to user identifications in the list.
<parameter name="service.user.allow" value="*USER"/>	The value *USER is a special case. It is a collective value for all users. In this example it allows access for all user identifications.
<parameter name="service.user.deny" value="*USER"/>	The value *USER is a special case. It is a collective value for all users. In this example it denies access to all user identifications.

The user authentication process looks for instances of the service.user.deny parameter and then instances of the service.user.allow parameter.

By default, LongReach denies access to user identifications beginning with the letter Q. To enable access for these user identifications, configure an explicit service.user.allow parameter.

Table 28 (page 55) illustrates a configuration that includes the service.user.allow and service.user.deny parameters.

Table 28: Example Configuration - User Allow and Deny Parameters

Controlling User Access - Configuration Example
<pre><?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs"></pre>

Controlling User Access - Configuration Example

```

index="index.html">
<errorlog enabled="true" file="www/instance/logs/error.log"/>
<accesslog enabled="true" file="www/instance/logs/access.log"/>
<listen secure="false" store="pki/wwwssl.jks" password="password"
port="6563"
sslport="6564"
interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1"
nodelay="false" timeout="5"/>
+<access>
+<mimetype>
<virtual host="*" active="true">
+<access>
+<protect>
<script>
  <match uri="/service/longreach.jsp"
    class="com.lansa.mobile.service.HTTPServiceLongReach"
    trace="true" clienttrace="false">
    <parameter name="service.default.host" value="LOCAL"/>
    <parameter name="service.remote.activation" value="
      7F04E051D3D9F04458D348832BE7CEFE15572A0F"/>
    <host name="LOCAL" system="LOCALHOST"/>
      <parameter name="service.access.log"
        value="www/instance/logs/longreach.log"/>
      <parameter name="service.user.deny" value="*USER"/>
    </host>
    <host name="CHICAGO" system="LOCALHOST"/>
      <parameter name="service.access.log"
        value="www/instance/logs/longreach.log"/>

```

<parameter name="service.user.allow" value="*USER"/>

```

<parameter name="service.folder.base"
  value="/longreachdata/user/{NAME}"/>
<parameter name="service.folder.shared"
  value="/longreachdata/shared"/>
<parameter name="service.folder.get.allow" value="{BASE}"/>
<parameter name="service.folder.get.allow" value="{BASE}/*"/>
<parameter name="service.folder.put.deny"
  value="{BASE}/manuals"/>
<parameter name="service.folder.put.deny"
  value="{BASE}/manuals/*">
<parameter name="service.folder.put.allow" value="{BASE}"/>
<parameter name="service.folder.put.allow" value="{BASE}/*"/>
<parameter name="service.folder.delete.deny"
  value="{BASE}/manuals"/>
<parameter name="service.folder.delete.deny"
  value="{BASE}/manuals/*">
<parameter name="service.folder.delete.allow" value="{BASE}"/>

```


Controlling User Access - Configuration Example

```

        <parameter name="service.folder.delete.allow" value="{BASE}/*">
        <notify data="{FOLDER}/{FILE}" key="{NAME}" padkey="true" ccsid="0"
            queue="/QSYS.LIB/LONGREACH.LIB/USER.DTAQ"/>
        <!--
        <notify data="{NAME},{FOLDER}/{FILE}" ccsid="0"
            queue="/QSYS.LIB/LONGREACH.LIB/FILE.DTAQ"/>
        -->
    </host>
</match>
</script>
+ <mimetype>
</virtual>
</instance>
</configuration>

```

Table 29 (page 57) provides examples and explanations for configuring the service.user.allow and service.user.deny parameters.

Table 29: User Allow and Deny Examples

Parameter	Explanation
<pre><parameter name= "service.user.allow" value= "*USER"/></pre>	This value allows every user identification, except for those beginning with the letter Q.
<pre><parameter name= "service.user.deny" value= "*USER"/></pre>	<p>This value denies access to every user, including those beginning with the letter Q.</p> <p>Using the *USER value on a service.user.deny parameter locks out every user. It overrides all values for the service.user.allow parameter.</p>
<pre><parameter name= "service.user.allow" value= "JOHNS,MARYB,JAMESD,WENDYF"/></pre>	This configuration allows all user identifications in the list.
<pre><parameter name="service.user.deny" value= "JOHNS,MARYB"/> <parameter name="service.user.allow" value= "*USER"/></pre>	<p>This configuration denies access to the users JohnS and MaryB, but allows all other users, except those beginning with the letter Q.</p> <p>This example illustrates the optimum method for allowing most users and denying a small number of users.</p>
<pre><parameter name="service.user.allow" value= "JOHNS,MARYB"/></pre>	This configuration allows only users JohnS and MaryB.

Parameter	Explanation
<code><parameter name="service.user.allow" value= "*USER,QSECOFR,QSYSOPR"/></code>	<p>This service.user.allow value allows all users and both QSECOFR and QSYSOPR.</p> <p>It is unnecessary to include service.user.deny parameters for user identifications beginning with the letter Q; LongReach denies access to these user identifications by default.</p>

Folder access permissions

Parameters explained in this section are:	service.folder.base service.folder.shared service.folder.get.allow service.folder.get.deny service.folder.put.allow service.folder.put.deny service.folder.delete.allow service.folder.delete.deny
---	---

Folder permissions apply to all users.

LongReach uses {BASE}, {NAME} and {SHARED} as substitution variables in the configuration file.

Base

The parameter service.folder.base defines the URI of the root folder containing user folders.

The {BASE} substitution variable represents this URI.

The {NAME} substitution variable represents the user identification as upper case.

The {name} substitution variable represents the user identification as lower case.

Shared

The parameter service.folder.shared defines the URI of the public shared folder.

The {SHARED} substitution variable represents this URI.

Folder get, put and delete

Get, put and delete are the actions available to users when working with folders. Get means copying or retrieving files from a folder. Put means placing files in a folder. Delete means removing or erasing files from a folder.

The service.folder parameter defines a folder or folders and the actions permitted. Table 30 (page 58) explains the syntax of folder parameters.

Table 30: Folder Get, Put and Delete Parameter Syntax

Parameter	Actions
<code><parameter name="service.folder.get.allow" value= "[folder]"/></code>	Users can get files from the folder defined in the value setting by [folder].

Parameter	Actions
<code><parameter name="service.folder.get.deny" value="[folder]"/></code>	Users are not permitted to get files from the folder defined in the value setting by [folder].
<code><parameter name="service.folder.put.allow" value="[folder]"/></code>	Users can put files into the folder defined in the value setting by [folder].
<code><parameter name="service.folder.put.deny" value="[folder]"/></code>	Users are not permitted to put files into the folder defined in the value setting by [folder].
<code><parameter name="service.folder.delete.allow" value="[folder]"/></code>	Users can delete files from the folder defined in the value setting by [folder].
<code><parameter name="service.folder.delete.deny" value="[folder]"/></code>	Users are not permitted to delete files from the folder defined in the value setting by [folder].

Table 31 (page 59) presents an example of a configuration file that includes folder parameters.

Table 31: Example Configuration – Folder Get Put and Delete Parameters

Controlling Access to Folders - Configuration Example
<pre> <?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> +<access> +<mimetype> <virtual host="*" active="true"> +<access> +<protect> <script> <match uri="/service/longreach.jsp" class="com.lansa.mobile.service.HTTPServiceLongReach" trace="true" clienttrace="false"> <parameter name="service.default.host" value="LOCAL"/> <parameter name="service.remote.activation" value=" 7F04E051D3D9F04458D348832BE7CEFE15572A0F"/> <host name="LOCAL" system="LOCALHOST"/> </pre>

Controlling Access to Folders - Configuration Example

```

<parameter name="service.access.log"
  value="www/instance/logs/longreach.log"/>
<parameter name="service.user.deny" value="*USER"/>
</host>
<host name="CHICAGO" system="LOCALHOST"/>
  <parameter name="service.access.log"
    value="www/instance/logs/longreach.log"/>
  <parameter name="service.user.allow" value="*USER"/>

  <parameter name="service.folder.base"
    value="/longreachdata/user/{NAME}"/>
  <parameter name="service.folder.shared"
    value="/longreachdata/shared"/>
  <parameter name="service.folder.get.allow" value="{BASE}"/>
  <parameter name="service.folder.get.allow" value="{BASE}/*"/>
  <parameter name="service.folder.put.deny"
    value="{BASE}/manuals"/>
  <parameter name="service.folder.put.deny"
    value="{BASE}/manuals/*">
  <parameter name="service.folder.put.allow" value="{BASE}"/>
  <parameter name="service.folder.put.allow" value="{BASE}/*"/>
  <parameter name="service.folder.delete.deny"
    value="{BASE}/manuals"/>
  <parameter name="service.folder.delete.deny"
    value="{BASE}/manuals/*">
  <parameter name="service.folder.delete.allow" value="{BASE}"/>
  <parameter name="service.folder.delete.allow" value="{BASE}/*">

  <notify data="{FOLDER}/{FILE}" key="{NAME}" padkey="true" ccsid="0"
    queue="/QSYS.LIB/LONGREACH.LIB/USER.DTAQ"/>
  <!--
  <notify data="{NAME},{FOLDER}/{FILE}" ccsid="0"
    queue="/QSYS.LIB/LONGREACH.LIB/FILE.DTAQ"/>
  -->
</host>
</match>
</script>
+ <mimetype>
</virtual>
</instance>
</configuration>

```

Table 32 (page 61) presents examples of actions permitted for a specific folder. In this example, the folder Manuals allows all users to get files from the folder, but put and delete actions are forbidden to all users. Placing files in the Manuals folder and its sub-folders requires action outside LongReach.

Table 32: Folder Get, Put and Delete Allow and Deny Examples – Manuals Folder

Parameter	Explanation
<parameter name="service.folder.get.allow" value="{BASE}/Manuals" />	All users may get files from the Manuals folder.
<parameter name="service.folder.get.allow" value="{BASE}/Manuals/*"/>	All users may get files from sub-folders in the Manuals folder.
<parameter name="service.folder.put.deny" value="{BASE}/Manuals" />	No user may put files into the Manuals folder.
<parameter name="service.folder.put.deny" value="{BASE}/Manuals/*"/>	No user may put files into sub-folder in the Manuals folder.
<parameter name="service.folder.delete.deny" value="{BASE}/Manuals" />	No user may delete files from the Manuals folder.
<parameter name="service.folder.delete.deny" value="{BASE}/Manuals/*"/>	No user may delete files from sub-folders in the Manuals folder.

Table 33 (page 61) is the configuration required to allow all users to get, put and delete files in the {BASE} folder and its sub-folders.

Table 33: Folder Get, Put and Delete Allow and Deny Examples – Base Folder

Parameter	Explanation
<parameter name="service.folder.get.allow" value="{BASE}" />	All users may get files from the {BASE} folder.
<parameter name="service.folder.get.allow" value="{BASE}/*"/>	All users may get files from sub-folders in the {BASE} folder.
<parameter name="service.folder.put.allow" value="{BASE}" />	All users may put files into the {BASE} folder.
<parameter name="service.folder.put.allow" value="{BASE}/*"/>	All users may put files into sub-folder in the {BASE} folder.
<parameter name="service.folder.delete.allow" value="{BASE}" />	All users may delete files from the {BASE} folder.
<parameter name="service.folder.delete.allow" value="{BASE}/*"/>	All users may delete files from sub-folders in the {BASE} folder.

Table 34 (page 61) is the configuration required to allow all users to get, put and delete files in the {SHARED} folder and its sub-folders.

Table 34: Folder Get, Put and Delete Allow and Deny Examples – SharedBox Folder

Parameter	Explanation
<parameter name="service.folder.get.allow" value="{SHARED}" />	All users may get files from the {SHARED} folder.
<parameter name="service.folder.get.allow" value="{SHARED}/*"/>	All users may get files from sub-folders in the {SHARED} folder.

Parameter	Explanation
<code><parameter name="service.folder.put.allow" value="{SHARED}" /></code>	All users may put files into the {SHARED} folder.
<code><parameter name="service.folder.put.allow" value="{SHARED}/*" /></code>	All users may put files into sub-folder in the {SHARED} folder.
<code><parameter name="service.folder.delete.allow" value="{SHARED}" /></code>	All users may delete files from the {SHARED} folder.
<code><parameter name="service.folder.delete.allow" value="{SHARED}/*" /></code>	All users may delete files from sub-folders in the {SHARED} folder.

To protect specific folders in SharedBox insert put.deny and delete.deny parameters for the folders.

File transfer notifications

Parameter explained in this section is: notify

To configure LongReach for file transfer notifications insert notify parameters in the configuration file and create one or more data queues. The notify parameter tells LongReach where to place the messages and also defines the format and content of a message. Refer to IBM documentation for an explanation of how to create data queues.

Table 35 (page 62) presents an example configuration including the notify parameter.

Table 35: Example Configuration – Notify Parameter

Notify - Configuration Example
<pre> <?xml version="1.0" encoding="UTF-8"?> <configuration> <instance name="HTTP Instance" active="true" root="www/instance/htdocs" index="index.html"> <errorlog enabled="true" file="www/instance/logs/error.log"/> <accesslog enabled="true" file="www/instance/logs/access.log"/> <listen secure="false" store="pki/wwwssl.jks" password="password" port="6563" sslport="6564" interface="*ALL" backlog="256" buffersend="-1" bufferreceive="-1" nodelay="false" timeout="5"/> +<access> +<mimetype> <virtual host="*" active="true"> +<access> +<protect> <script> <match uri="/service/longreach.jsp" class="com.lansa.mobile.service.HTTPServiceLongReach" trace="true" clienttrace="false"> </pre>

Notify - Configuration Example

```

<parameter name="service.default.host" value="LOCAL"/>
<parameter name="service.remote.activation" value="
7F04E051D3D9F04458D348832BE7CEFE15572A0F"/>
<host name="LOCAL" system="LOCALHOST"/>
  <parameter name="service.access.log"
    value="www/instance/logs/longreach.log"/>
  <parameter name="service.user.deny" value="*USER"/>
</host>
<host name="CHICAGO" system="LOCALHOST"/>
  <parameter name="service.access.log"
    value="www/instance/logs/longreach.log"/>
  <parameter name="service.user.allow" value="*USER"/>
  <parameter name="service.folder.base"
    value="/longreachdata/user/{NAME}"/>
  <parameter name="service.folder.shared"
    value="/longreachdata/shared"/>
  <parameter name="service.folder.get.allow" value="{BASE}"/>
  <parameter name="service.folder.get.allow" value="{BASE}/*"/>
  <parameter name="service.folder.put.deny"
    value="{BASE}/manuals"/>
  <parameter name="service.folder.put.deny"
    value="{BASE}/manuals/*">
  <parameter name="service.folder.put.allow" value="{BASE}"/>
  <parameter name="service.folder.put.allow" value="{BASE}/*"/>
  <parameter name="service.folder.delete.deny"
    value="{BASE}/manuals"/>
  <parameter name="service.folder.delete.deny"
    value="{BASE}/manuals/*">
  <parameter name="service.folder.delete.allow" value="{BASE}"/>
  <parameter name="service.folder.delete.allow" value="{BASE}/*"/>

  <notify data="{FOLDER}/{FILE}" key="{NAME}" padkey="true"
    ccsid="0" queue="/QSYS.LIB/LONGREACH.LIB/USER.DTAQ"/>
  <!--
  <notify data="{NAME},{FOLDER}/{FILE}" ccsid="0"
    queue="/QSYS.LIB/LONGREACH.LIB/FILE.DTAQ"/>
  -->

</host>
</match>
</script>
+ <mimetype>
</virtual>
</instance>
</configuration>

```

The notify parameter has data, key and queue properties. Data defines the message content. Key describes the keyed property of the data queue and the data used as the key. Queue defines the data queue name and the library in which the data queue resides.

LongReach allows only one active notify parameter. The configuration file may contain multiple notify parameters but LongReach will choose the last notify parameter. The example configuration file shows the second notify parameter enclosed by `<!-- -->` (as a comment).

LongReach uses the following substitution variables in notify parameters.

{FILE}	The file variable represents the name of the transferred file.
{FOLDER}	The folder variable represents the name of the folder in which the file resides.
{NAME}	The upper case NAME variable represents user identification as upper case.
{name}	The lower case name variable represents user identification as lower case.

Table 36 (page 64) presents two examples of the notify parameter.

Table 36: Notify Parameter Definition and Examples

Parameter	Explanation
<code><notify data="{FOLDER}/{FILE}" key="{NAME}" queue="/QSYS.LIB/LONGREACH.LIB/USER.DTAQ" padkey="true" ccsid="0"/></code>	<p>LongReach will place messages in a queue named USER which resides in the folder defined by queue. The key for this queue is the name.</p> <p>The message content is folder name, a forward slash and file name.</p>
<code><notify data="{NAME},{FOLDER}/{FILE}" queue="/QSYS.LIB/LONGREACH.LIB/FILE.DTAQ" ccsid="0"/></code>	<p>LongReach will place messages in a queue named FILE which resides in the folder defined by queue. This queue has no key.</p> <p>The message content is user name, a comma, folder name, a forward slash and file name.</p>

You will need a program to monitor the queues, extract the messages and either process the message or pass the message to another program.

Configure multiple instances and virtual servers

Anatomy of a configuration file

A LongReach server configuration file contains two main sections:

- instance (`<instance> ... </instance>`), and
- virtual (`<virtual> ... </virtual>`) sections.

An instance represents a HTTP server and a virtual represents a virtual server within the HTTP server.



A configuration contains one or more instances.

An instance contains one or more virtual servers.

Figure 17 (page 65) shows a configuration of one instance and one virtual server.

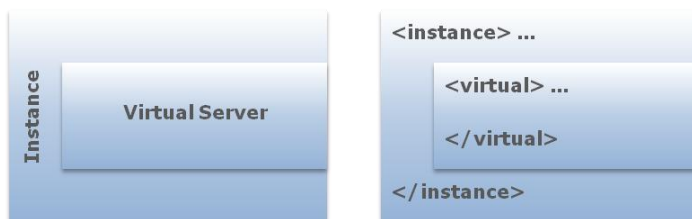


Figure 17: Configuration of a Single Instance and a Single Virtual Server

Figure 18 (page 65) shows a configuration of one instance and multiple virtual servers.

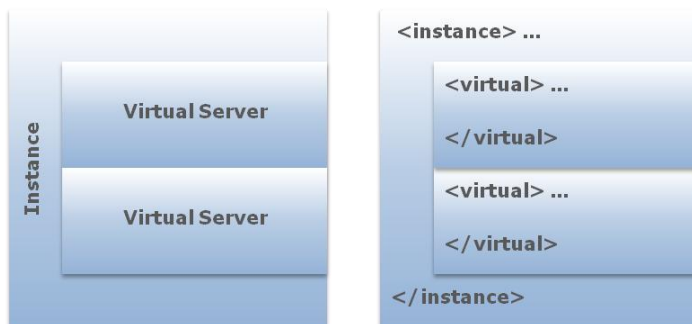


Figure 18: Configuration of a Single Instance with Multiple Virtual Servers

Figure 19 (page 65) shows a configuration of multiple instances and multiple virtual servers.

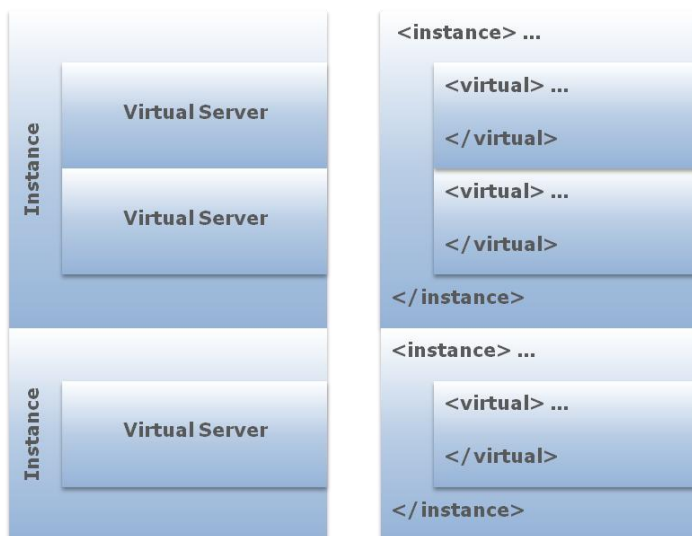


Figure 19: Configuration of Multiple Instances with Multiple Virtual Servers

Multiple HTTP server instances

To configure multiple HTTP server instances (`<instance> ... </instance>`), copy the default instance for each required server and then configure the directives applicable to each instance.

Multiple virtual servers

To configure multiple virtual servers (`<virtual> ... </virtual>`) in an instance, copy the default virtual server for each required virtual server and then configure the directives applicable to each virtual server.

LongReach server configuration reference guide

The reference guide provides explanations of the individual items in the httpd configuration file. The configuration item reference section is a dictionary containing definitions of the individual configuration settings. The sections describing MIME types and allow/deny directives provide information about these topics.

Configuration item reference

Table 37 (page 66) provides explanations of configuration items in the JSM HTTP Server httpd configuration file. The items in the table are in name sequence. The configuration item name is the name and position of the parameters in the httpd configuration file.

Table 37: Server Reference: Configuration Item Reference

Configuration Items	Definitions
Access allow	List of addresses, content lengths and/or user agents allowed access to the JSM HTTP Server. Example: <allow address="10.2.1.45">
Access deny	List of addresses, content lengths and/or user agents denied access to the JSM HTTP Server. Example: <deny address="10.2.1.45">
Access log enabled	LongReach logs all access events when access log enabled is "true". LongReach logs no access events when access log enabled is "false".
Access log file	Name and location of the access log file. Example: "www/instance/logs/access.log"
Cache maxage	Maximum age for cached files in seconds.
Cache maxage image	Maximum age for cached image files in seconds.
Cache maxage pdf	Maximum age for cached PDF files in seconds.
Error log enabled	LongReach logs all errors when error log enabled is "true". LongReach logs no errors when error log enabled is "false".
Error log file	Name and location of the error log file. Example: "www/instance/logs/error.log"
Host directive	The host directive defines one or more servers that LongReach will use and the host definition includes the directives and parameters that influence LongReach server behaviour.
Instance active	To operate LongReach this parameter will always be "true". The JSM HTTP Server supports multiple instances and needs this parameter to indicate which instances to activate at run time.

Configuration Items	Definitions
Instance index	Name of the index page. Example: "index.html"
Instance name	Name allocated the JSM HTTP Server. Examples: "WebServer" or "HTTP Instance" This name does not need to be the same as the instance name in the manager.properties file.
Instance root	Name of the directory from which the documents will be served by the JSM HTTP Server. Also known as document root. Example: "www/instance/htdocs"
Listen backlog	The backlog defines the depth of the TCP/IP queue.
Listen buffer receive	TCP/IP receive-buffer size in bytes. Special case value "-1" means use operating system default.
Listen buffer send	TCP/IP send-buffer size in bytes. Special case value "-1" means use operating system default.
Listen interface	TCP/IP interface address that the JSM HTTP Server will bind to and accept connections on. Default value is *ALL *ALL will bind to all interfaces on the server.
Listen no delay	True enables TCP/IP no delay option. False means use the operating system setting for this parameter.
Listen password	Password that will open the store file used for TLS/SSL configuration. Example: "password"
Listen port	TCP/IP port number the server will use to accept connections on.
Listen secure	When true, the JSM HTTP Server uses TLS/SSL When false, the JSM HTTP Server uses plain sockets.
Listen sslport	TCP/IP port number the server will use to accept connections when using TLS/SSL.
Listen store	Path for the store file that contains the private key and public certificates (used when TLS/SSL enabled). Fully qualified name and location of the store file. Example: "pki/wwwssl.jks"

Configuration Items	Definitions
Listen timeout	Time out count in seconds (integer).
Mimetype map extension	File extension used to identify the MIME type. Example: "png"
Mimetype map type	The type describes the nature of the MIME type. Examples: "image/png"
Notify	Defines the message queue(s) LongReach will use for file transfer notifications.
Protect match authentication	Type of authentication, values are: "basic", "digest" or "basic,digest". Basic: Basic authentication is a concatenation of user name, a colon and the password encoded with the Base64 algorithm. Digest: Digest authentication is an application of MD5 cryptographic hashing of user credentials. It provides stronger encoding than basic authentication.
Protect match realm	Name of the realm used for authentication to parts of the Web site or application. The realm is associated with the matched URIs.
Protect match uri	URIs used to match against requests. When a match occurs the associated realm provides the authentication details.
Protect realm name	Name assigned to a realm.
Protect realm user access	Access is a digest of user information (including the password).
Protect realm user name	User name for authentication.
Remote activation key	License key that allows access to remote servers where the LongReach server software is not installed.
Script match	The match definitions describe services that the JSM HTTP Server will use depending on the match criteria in the URI.
Script match class	Class is the name of a service.
Script match client trace	Use this parameter to trace activity associated with a client. When "true" tracing will occur for clients that append ?trace=true to the URL. The value "false" disables client-initiated tracing. Tracing will slow performance.

Configuration Items	Definitions
Script match trace	Use this parameter to trace activity associated with all clients. Broader scope than client trace. The value "true" enables tracing. The value "false" disables tracing.
Script match URI	The match URI is the match criteria the JSM HTTP Server uses to determine the services to use.
service.access.log	Defines the location of the log file LongReach will use to log access events.
service.default.host	Defines the host server.
service.folder.base	Defines the root folder for files and folders managed by LongReach.
service.folder.delete.allow	Allows users to delete files from folders.
service.folder.delete.deny	Prevents users from deleting files from folders.
service.folder.get.allow	Allows users to get files from folders.
service.folder.get.deny	Prevents users from getting files from folders.
service.folder.put.allow	Allows users to put files in folders.
service.folder.put.deny	Prevents users from putting files in folders.
service.folder.shared	Defines the physical folder used for SharedBox.
service.remote.activation	This is the LongReach license key to unlock the services that access remote servers. Using remote activation, LongReach can access files and folders on servers where LongReach server is not installed.
service.user.allow	Allows users to access a LongReach server.
service.user.deny	Prevents users from accessing a LongReach server.
Virtual access allow	List of addresses, content lengths and/or user agents allowed access to the JSM HTTP Server. Example: <allow useragent="safari">
Virtual access deny	List of addresses, content lengths and/or user agents denied access to the JSM HTTP Server. Example: <deny useragent="safari">
Virtual active	When the value is "true" this virtual host is active. When the value is "false" the virtual host is inactive. To operate LongReach the value must be "true". Services configured in the virtual host are unavailable when the virtual host is inactive.

Configuration Items	Definitions
Virtual host	<p>Name of the virtual host to match with the HTTP host property. This allows multi homing. HTTP requests can be directed to different virtual host sections of the configuration in the server instance.</p> <p>If a virtual host is not found then the connection request is rejected.</p> <p>The special case value "*" accepts requests from any HTTP host. Specific names take precedence.</p>
Virtual mimetype map extension	<p>File extension used to identify the MIME type.</p> <p>Example: "png"</p>
Virtual mimetype map type	<p>The type describes the nature of the MIME type.</p> <p>Examples: "image/png"</p>

About MIME types

MIME type describes the nature of content of file for the JSM HTTP Server. Table 38 (page 70) provides the default list of MIME types supported by LongReach.

Table 38: Server Reference: MIME Types

Segment	Extension	Names
Audio/Video	wav	audio/wav
Audio/Video	mpa	audio/x-mp4
Audio/Video	mp4	video/mp4
Audio/Video	mov	video/quicktime
Audio/Video	ts	video/MP2T
Audio/Video	m3u8	application/x-mpegURL
General	zip	application/zip
General	zipx	application/zip
General	pdf	application/pdf
General	txt	text/plain; charset=utf-8
HTML	css	text/css; charset=utf-8
HTML	xsl	text/xls; charset=utf-8
HTML	xml	text/xml; charset=utf-8
HTML	htm	text/html; charset=utf-8
HTML	html	text/html; charset=utf-8
HTML	js	application/x-javascript; charset=utf-8

Segment	Extension	Names
HTML	appcache	text/cache-manifest; charset=utf-8
HTML	manifest	text/cache-manifest; charset=utf-8
Images	png	image/png
Images	gif	image/gif
Images	jpg	image/jpeg
Images	jpeg	image/jpeg
Images	tiff	image/tiff
Images	ico	image/x-icon
Images	svg	image/svg+xml
iWork	key	application/vnd.apple.keynote
iWork	pages	application/vnd.apple.pages
iWork	numbers	application/vnd.apple.numbers
Office	doc	application/msword
Office	dot	application/msword
Office	docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
Office	dotx	application/vnd.openxmlformats-officedocument.wordprocessingml.template
Office	docm	application/vnd.ms-word.document.macroEnabled.12
Office	dotm	application/vnd.ms-word.template.macroEnabled.12
Office	xls	application/vnd.ms-excel
Office	xlt	application/vnd.ms-excel
Office	xla	application/vnd.ms-excel
Office	xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
Office	xltx	application/vnd.openxmlformats-officedocument.spreadsheetml.template
Office	xlsm	application/vnd.ms-excel.sheet.macroEnabled.12
Office	xltm	application/vnd.ms-excel.template.macroEnabled.12
Office	xlam	application/vnd.ms-excel.addin.macroEnabled.12
Office	xlsb	application/vnd.ms-excel.sheet.binary.macroEnabled.12
Office	ppt	application/vnd.ms-powerpoint

Segment	Extension	Names
Office	pot	application/vnd.ms-powerpoint
Office	pps	application/vnd.ms-powerpoint
Office	ppa	application/vnd.ms-powerpoint
Office	pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation
Office	potx	application/vnd.openxmlformats-officedocument.presentationml.template
Office	ppsx	application/vnd.openxmlformats-officedocument.presentationml.slideshow
Office	ppam	application/vnd.ms-powerpoint.addin.macroEnabled.12
Office	pptm	application/vnd.ms-powerpoint.presentation.macroEnabled.12
Office	potm	application/vnd.ms-powerpoint.template.macroEnabled.12
Office	ppsm	application/vnd.ms-powerpoint.slideshow.macroEnabled.12
PKI	crl	application/x-pkcs7-crl
PKI	crt	application/x-x509-ca-cert

Address allow and deny directives syntax

Table 39 (page 72) shows examples of the access allow/deny directive for addresses.

Table 39: Server Reference: Access Allow and Deny Addresses

Allow/Deny	Syntax and Examples
Allow any address	<allow address="*">
Allow addresses in a range	<allow address="10.2.1">
Allow a specific address	<allow address="10.2.1.45">
Deny any address	<deny address="*">
Deny addresses in a range	<deny address="10.2.1">
Deny a specific address	<deny address="10.2.1.45">

Table 40 (page 72) shows examples of the access allow/deny directive for content length.

Table 40: Server Reference: Access Allow and Deny Content Length

Allow/Deny	Syntax and Examples
Allow access for content less than or equal to the specified length	<allow contentlength="4096">

Allow/Deny	Syntax and Examples
Zero content length is a special case to allow access for no content connections from the browser	<allow contentlength="0">
Deny access for content greater than the specified length	<deny contentlength="4096">

User agents are applications or services that act on behalf of the user. When a user requests a web page (or URL), the browser acts as a user agent by sending the page request to the JSM HTTP Server. Examples of user agents are browsers, web crawlers, link checkers, bots and email clients. Access allow and deny directives control which user agents the JSM HTTP Server will allow or deny access.

Table 41 (page 73) shows examples of user agents and the syntax of the allow access and deny access directives.

Table 41: Server Reference: Access Allow and Deny User Agents

Allow/Deny	Syntax and Examples
Allow access for any user agent	<allow useragent="*"
Allow access if no user agent provided	<allow useragent="?"
Allow access to Chrome	<allow useragent="chrome"
Allow access to the Internet Explorer	<allow useragent="explorer"
Allow access to Firefox	<allow useragent="firefox"
Allow access to Safari	<allow useragent="safari"
Deny access for any user agent	<deny useragent="*"
Deny access if no user agent provided	<deny useragent="?"
Deny access to Chrome	<deny useragent="chrome"
Deny access to Internet Explorer	<deny useragent="explorer"
Deny access to Firefox	<deny useragent="firefox"
Deny access to Safari	<deny useragent="safari"

The evaluation of the directives starts with the first item in the list and continues until it finds a true condition. Any combinations of address, user agent and content length are acceptable. However, it is possible to negate the effect of a directive by its position in the list. For example, placing an allow any user agent (<allow useragent="*" />) ahead of a deny for a specific user agent (<deny useragent="opera" />) will cause the JSM HTTP Server to ignore the deny directive.

Table 42 (page 74) presents lists of user agents.

Table 42: Server Reference: Sample Lists of User Agents

Browser User Agents	Bots and Device User Agents
android	googlebot
chrome	googletoolbar
explorer	imac
firefox	lansaua
opera	longreach
safari	msnbot
safari (iPad)	webdavnav
safari (iPhone)	yahoobot
safari (iPod)	

Appendices

Abbreviations and Terms

Table 43 (page 75) presents definitions for abbreviations and terms used in this guide.

Table 43: Abbreviations and Terms

Abbreviations and Terms	Definitions and Explanations
CCSID	Coded Character Set Identifier
DBCS	Double Byte Character Set
Directives	Configuration directives are the parameters and settings that control the behaviour of the JSM HTTP Server.
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IFS	The Integrated File System is a part of the IBM i operating system. It supports stream input/output and storage management capabilities like personal computer and UNIX operating systems.
Internet Media Type	Internet Media Type (IMT) is an identifier for file formats on the Internet. An example is the IMT for the Portable Document Format: application/pdf See also MIME type.
JSM	Java Services Manager
LongReach server	This collective term refers to the LongReach components that reside on the server.
MIME	Multipurpose Internet Mail Extensions MIME is an Internet standard that describes content. An example is PNG for Portable Network Graphic files. See also Internet Media Types.
Realm	A realm is a database containing a list of valid users of a web application. The user information in the database is user name, password and a set of roles associated with the user.
SBCS	Single Byte Character Set
SSL	Secure Sockets Layer
System directory	System directory is a synonym for system folder.
System folder	The system folder location is: longreach/jsm/instance/system
TCP/IP	Transmission Control Protocol/Internet Protocol

Abbreviations and Terms	Definitions and Explanations
TLS	Transport Layer Security

Assumed and prerequisite knowledge

This guide assumes knowledge of the following topics.

Table 44: Assumed and Prerequisite Knowledge

Subject Matter	Explanations
XML files	The directives for LongReach reside in the configuration file, formatted as XML. You need to be able to read and edit the XML.

Mobile device requirements

This section describes prerequisites and system requirements for running LongReach on a mobile device.

Hardware and operating system

LongReach operates on iPhone and iPad devices. It has been tested successfully on the following device and operating system combinations.

Devices	iOS Versions
iPad 2	iOS 4.3.5
	iOS 5.0
	iOS 5.0.1
iPhone 3G	iOS 4.2.1
iPhone 3G S	iOS 4.1
iPhone 4	iOS 4.2.1
	iOS 4.3.5
	iOS 5.0
	iOS 5.0.1
iPhone 4 S	iOS 5.0
	iOS 5.0.1

Software

The Long Reach app must be installed on the mobile device. It is available for download from the Apple Store (iTunes).

Connectivity

To use LongReach with a server configure the communications settings on the mobile device. LongReach uses communications service available on a mobile device including Wi-Fi and cell phone networks.

What ports does LongReach use?

The LongReach server uses four ports:

6560	LongReach Manager	This port is for internal use only; do not open for public access.
6561	LongReach Console	This port is for internal use only; do not open for public access.
6563	LongReach HTTP Server	This is the default port used when connecting to LongReach via HTTP. The port number is configurable.
6564	LongReach HTTPS Server	This is the default port used when connecting to LongReach via HTTPS. The port number is configurable and is only needed when using TLS/SSL.

LongReach Manager and Console

The LongReach server binds to ports 6560 and 6561; but does not accept TCP/IP connections on these ports. The manager.properties configuration file includes keywords to control LongReach accepting TCP/IP connections on these ports.

```
tcp.client.address=*none
```

```
studio.client.address=*none
```

```
console.client.address=*none
```

Setting the values of these keyword to *none prevents LongReach accepting TCP/IP connections on these ports.

LongReach HTTP and HTTPS ports

LongReach requires only one port open for public access. This port is either the HTTP port or the HTTPS port and the HTTPS port is not active in the default LongReach configuration.

LongReach supports concurrent instances HTTP and HTTPS and in this case both ports must be active.

Both the HTTP and HTTPS ports are configurable and administrators can choose the port numbers for LongReach to use.

Using ports 80 and 443

Administrators can use port 80 for HTTP and port 443 for HTTPS instead of the default ports numbers in the LongReach configuration, provided that these ports are not already in use on the server that hosts LongReach.

Proxy server

The LongReach server can be used with a proxy server. In this case the mobile device will connect to the proxy server using an appropriate port number and the proxy server will in turn connect to the LongReach server on a default port (for example 6563 got HTTP).

The advantage of using a proxy server is that the LongReach server is hidden from public view and installing LongReach requires no additional Internet-facing ports on the proxy server.

Data protection in the LongReach app

How LongReach data protection works

Apple's Data Protection provides data security and LongReach takes advantage of iOS's Data Protection API. This API encrypts files, so that the files will be secure, even if the mobile device is lost or stolen.

Data Protection works in conjunction with the passcode that you use to unlock your mobile device and is active only if you enable Passcode Lock. If you don't lock your device with a passcode, your files will not be encrypted. However, locking a device with a passcode doesn't automatically mean that the files will be encrypted.

Data protection is only as good as your passcode. Choose strong alphanumeric passwords consisting of random characters, not words and don't forget to change your passwords periodically.

All files managed by LongReach are enabled for iOS Data Protection. LongReach itself doesn't encrypt or decrypt the files; iOS encrypts and decrypts the files. When you unlock your mobile device with a passcode, all files can be decrypted and fully accessible, even if LongReach app is not being run at the time. Always lock your device with a passcode.

Apple iOS security hardening checklist

The following items are actions users of iOS devices can take to improve the level of security on their devices.

Data protection enabled
Enable locks for password, cover-lock and auto-lock
Erase data before return, repair or recycling the mobile device
Require a passcode
Require password immediate
Set auto-lock timeout
Turn off Ask to Join Networks
Turn off Bluetooth

Using EBCDIC files in the LongReach app

The potential for file rendering errors in LongReach is limited to files created by software, or copied by a person, in the IFS on an IBM i server that are not encoded as UTF-8 and CCSID 1208. File rendering errors will occur only when the files are transferred to a mobile device and no viewer or application is available to read the files.

To view or edit files on a mobile device you need viewer software, an editor or an application that can read and interpret the file content. To view a photo the LongReach app opens the photo software on the mobile device; it does not read the file content and render the photo image. Safari is the default viewer software when no viewer or application is available.

LongReach treats files as binary. Therefore, a file on the server is the same as the copy of the file on a mobile device. No encoding conversions occur for files such as images, photos, PDFs, video recordings, audio recordings or Microsoft Word documents.

UTF-8 and CCSID 1208 is the encoding LongReach applies to all files it transfers from a mobile device to a LongReach server. This means that programs on the server can read and interpret the content of these files and the IBM i operating system takes care of encoding conversions.

LongReach is not concerned with file content. The text editor in LongReach is the one exception, and in this case LongReach is the viewing and editing application. Creating a text file using the LongReach app and transferring it to the LongReach server will not cause encoding errors as LongReach encodes text files it creates or edits as UTF-8 with a CCSID of 1208.

Possible causes of encoding errors are files created by software on IBM i servers. LongReach regards EBCDIC files without a .txt suffix as binary and the file will remain unchanged when transferred to the LongReach app on a mobile device. If no application is available as the file viewer, LongReach will use Safari to render the file content, and of course what you see is most likely nonsense. The LongReach text editor will open EBCDIC files with a .txt suffix and try to render the content which is also likely to be nonsense.

The simplest way to use an EBCDIC file with LongReach is to copy the original file, add a .txt suffix and set the encoding as UTF-8 and CCSID 1208. The copy command includes parameters to specify encoding as UTF-8 and CCSID 1208. This method removes the need to amend any software that creates the files and also ensures that the file content is understandable on a mobile device. Alternatively, you could change the software that creates the files to use UTF-8 and CCSID 1208.

Why can't LongReach convert EBCDIC files automatically? LongReach is not concerned with file content, except for text files it creates and these are encoded correctly. Therefore, LongReach does not need to convert any files. If LongReach does not need to convert file, how would it know which files to convert? The .txt suffix might be an indicator, but then LongReach has to interrogate the encoding properties of each .txt suffixed file to find files that are not UTF-8 and CCSID 1208 and then copy or convert the files. Converting every file with a .txt suffix for every file transfer to a mobile device will generate additional work and slow the file transfer.

File and folder access permissions

LongReach server uses object access control services provided by the IBM i server operating system to manage access authority to files and folders. When users create files and folders on the server the files and folders inherit access authorities from their parent folder. Administrators can change the authorities, for example to exclude public access or allow read only access by other user profiles.

The top level folder for the LongReach user JOHND is /longreachdata/user/JOHND and he owns the folder, and also any files and sub-folders in this top folder.

Administrators can grant access to specific files and folders using the LongReach server directive service.folder.get.allow. For example, administrators can insert the directive service.folder.get.allow for the sub-folder /longreachdata/user/JOHND/Marketing and any user can retrieve files from the folder.

If an administrator changes the operating system access permissions on the folder /longreachdata/user/JOHND/Marketing to exclude public access, only the user JOHND can access the folder.

The operating system permissions always take precedence over LongReach server directives.