*First*PAGE – v9.19
Protocol Specification

SeQent
Suite 137
4500 Blakie Road
London, Ontario
Canada
N6L 1G5

Terms of Use

Information in this document is subject to change without notice and does not represent a commitment on the part of SeQent. The software, which includes information contained in any databases, described in this document is furnished under a license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of SeQent.

This software program and associated documentation are provided in an "as is" condition. In no event shall SeQent, its employees, directors or agents be liable for any loss or damage whatsoever (including, but not limited to, damages for loss of business profits, business interruption, loss of business information, or other monetary loss) arising out of the use of or the inability to use this software. By installing this software on your computer system you acknowledge that you have read and understood these terms and that you accept them.

Printed in Canada

Document Date: May 28, 2014

Edits

Version	Date	Author	Description
9.00	September 1, 2008	S. Burns, C. Longman, D. Vlasman	Initial prototype Added DESTINATION field for USER;DETAIL;
9.01	December 1, 2009	S. Burns, D. Vlasman	Expanded USER;DETAIL; call to support admin lookup of other user details.
9.02	January 28, 2011	S. Burns	Corrected USER;DETAIL; items to return dests and dist lists.
9.03	February 4, 2011	J. Weber	Added new protocol item USER;DESTLIST; and removed the destination details from the USER;DETAIL; call
9.04	April 26, 2011	S. Burns	Added new protocol items PROV;FAILOVER; and PROV;FAILBACK;
9.05	May 3, 2011	J. Weber	Added new PROV;STATE; command
9.07	Aug 12, 2011	D. Vlasman	Add new Windows Trusted security mode
9.08	July 4, 2012	S. Burns	Add USER;TYPE; command
9.09	August 4, 2012	J. Weber	New PROV;TRANSCOUNT;
9.10	April 25, 2013	S. Burns	Deprecated PAGE;QUEUE; Defined DEST;QUEUE;
9.11	May 23, 2013	S. Burns	Clarify result packets. Clarify recurrence items
9.12	June 24, 2013	S. Burns	Clarify example session and dest calls
9.13	July 4, 2013	J. Weber	Clarify DEST;QUEUE; injection
9.15	December 13, 2013	S. Burns	Clarify TRANS;DETAIL;
9.16	April 24, 2014	E. Jones	Added GROUPLIST field for USER;DETAIL;
9.17	May 11, 2014	S. Burns	Clarify SESSION;WRITE;RETRIES;

9.18	May 21, 2014	J. Weber	Corrected SESSION;WRITE;RETRIES description.
9.19	May 28, 2014	S. Burns	Clarified in-secure server login account case.

Table of Contents

Chapter One
Protocol Overview

This chapter introduces ***First*PAGE** protocol and describes how to utilize it from within your applications.

Refer to the *First*PAGE Server Installation and User manual for more details on the *First*PAGE Server product.

Getting Started

This manual contains procedural and conceptual information about the ***First*PAGE Protocol.** It is written for System Administrators and Developers to enable them to utilize *First*PAGE from within their applications.

While it is not required that the user have any prior programming or alarm management experience, the user should understand basic Windows concepts.

Documentation Conventions

This document contains the following documentation conventions to help you navigate through the manual, obtaining a better understanding of the material.

- Notes contain tips or reminders about procedural and conceptual information within the manual.

- Cross-references provide you with a link to further information about the section of the document that you are currently reading.

Chapter Two
*First*PAGE Server

This chapter describes ***First*PAGE Server** fundamentals that the application developer should know before attempting to develop a *First*PAGE enabled application.

Server Engine

*First*PAGE Server is a Windows service that provides an advanced interface for managing wireless message notifications. To interface with the engine applications, indicate which host they are running on, which application they are and what alarm from their environment is now active or inactive.

This document will describe these items in detail so that the developer may form the correct packets to achieve the desired results.

Connecting to the Server

*First*PAGE Server listens, by default, on TCP/IP port number 8003. To connect to the engine the user must open a TCP/IP socket to the appropriate IP host, using this port. After this connection is completed the USER;LOGIN; operation must be performed. Once the connection is open the user may begin to execute the commands indicated below. You may opt to maintain this connection or it may be open and closed for each transaction if the application is not robust enough to provide reconnect on loss of this connection.

Disconnecting from the Server

When the application is finished with the *First*PAGE Server any open connections to the server should first be closed.

Object Terminology

Within this document the following acronyms (as detailed in the glossary) will be utilized.

Term	Description
Destination	Message end point on a service provider (ie. a mobile handset)
Provider	Messaging service provider (ie. mobile carrier, in-house paging system etc.)

Packet Formats

All packets sent to and from the *First*PAGE Server follow a standard format. Note that STATUS results MUST be read from the socket before performing the next operation. If results are provided they must also always be read. A standard transaction follows the following convention:

Client: operation;sub-op;[data;]<cr>

Server: operation;STATUS;code;message;<cr>

Field	Description	Notes
Operation	The operation family being requested.	
sub-op	The sub-operation within the operation family desired.	
Data	Any optional data required for the operation/sub-op to	

	be executed.	
STATUS	A static value always included in a 'STATUS' packet to indicate its type.	
Code	The status code resulting from the execution of the operation/sub-option.	0 = Success Non-0 = Error/Failure
Message	The status message explaining the reason for the status code.	
<cr>	ASCII Dec. 13	

For extended status commands the format above is consistent but one or more result packets may follow. Result packets will only be returned for commands that need to return more data to the client, than just a status code. **Results will only be returned if the 'code' value of the STATUS packet is 0, which indicates the command was a success.** The format of an extended transaction is as follows:

Client: operation;sub-op;[data;]<cr>

Server: operation;STATUS;0;message;<cr>

Server: operation;RESULT;eofflag;data;<cr>

For the 'RESULT' packet, the 'eofflag' will be '0' if another result packet will follow, or '1' if this is the last result packet. The data portion of the packet is specific to the command being executed, and the trailing ';' is always provided. All results MUST be read until the RESULT;1; packet is received until the final <cr> character.

Field	Description	Notes

RESULT	A static value always included in a 'RESULT' packet to indicate its type.	
eofflag	End of File flag. 1 if this is the last RESULT packet, 0 if more will follow.	
data	Operation/Sub-Op specific data.	
<cr>	ASCII Dec. 13	

Note: The ';' character, as it is used as the protocol field separator, must be escaped if it is used within a field. To accomplish this prefix any ';' character with a '\' character, and escape any '\' character with another '\' character.

Chapter Three
ADMIN Operations

This chapter describes the ***First*PAGE** protocol subset known as the ADMIN operations. These commands, typically only used by administration tools, allow the developer to request information about the ***First*PAGE** Server status, force engine reload. These commands are not normally referenced by ***First*PAGE** gateways.

DETAIL

The DETAIL sub-operation requests that the engine return certain details to allow the administrator to determine which version is running.

Format: ADMIN;DETAIL;
Status: ADMIN;STATUS;statuscode;statusmessage;
Result: ADMIN;RESULT;1;FirstPAGE Server|hostname|major_ver|complete_ver|platform|start_datetime;

OUTPUT	
Field	**Description**
Server Name	Always "*First*PAGE Server"
hostname	Host name of the server running the server
major_ver	Major version #
complete_ver	Complete version specification
platform	Platform specification
start_datetime	Date/time the engine started specified in UCT

RELOAD

The RELOAD sub-operation indicates to the *First*PAGE Server that it should reload all internal tables, most likely because its database contents have been manipulated and it needs to make changes to its internal data structures as appropriate.

Format: ADMIN;RELOAD;
Status: ADMIN;STATUS;statuscode;statusmessage;

SECMODE

The SECMODE sub-operation requests that the engine return details as to which security mode it is running under.

Format: ADMIN;SECMODE;
Status: ADMIN;STATUS;statuscode;statusmessage;
Result: ADMIN;RESULT;1;Secmode;

OUTPUT	
Field	**Description**
Secmode	Security mode bitmask. Bit: 0 – Authentication required Off = No authentication required On = Authentication required Bit: 1-4 - Unused Bit: 5 – Ford Authentication (Requires bit 0 be on) Off = Ford authentication off On = Ford authentication on Bit: 6 – Chrysler Authentication (Requires bit 0 be on) Off = Chrysler authentication off On = Chrysler authentication on Bit: 7 – Windows Trusted Authentication (Requires bit 0 be on) Off = Windows Trusted authentication off On = Windows Trusted authentication on

SHUTDOWN

The SHUTDOWN sub-operation requests that the engine cleanly shutdown. The engine will proceed cleanly disconnect from all providers and log any unsent messages.

Format: ADMIN;SHUTDOWN;

Status: ADMIN;STATUS;statuscode;statusmessage;

Chapter Four
ALIAS Operations

This chapter describes the ***First*PAGE** protocol subset known as the ALIAS operations. These commands allow the developer to request alias information from the *First*PAGE Server.

LIST

The LIST sub-operation facilitates the interrogation of the known aliases.

Format: ALIAS;LIST;
Status: ALIAS;STATUS;statuscode;statusmessage;
Result: ALIAS;RESULT;0;alias1_a;alias1_b;alias1_c;alias1_locator;
Result: ALIAS;RESULT;0;alias2_a;alias2_b;alias2_c;alias2_locator;
Result:
ALIAS;RESULT;1;aliasN_a;aliasN_b;aliasN_c;aliasN_locator;

RELOAD

The RELOAD sub-operation indicates to the *First*PAGE Server that it should reload all internal tables related to aliases.

Format: ALIAS;RELOAD;
Status: ALIAS;STATUS;statuscode;statusmessage;

Chapter Five
ATTRIB Operations

This chapter describes the ***First*PAGE** protocol subset known as the ATTRIB operations. These commands, typically only used by administration tools, allow the developer to request information about the ***First*PAGE** Server. These commands are not normally referenced by ***First*PAGE** gateways.

LIST

The LIST sub-operation requests that the engine return certain details to allow the administrator to determine which version is running. The parameter following list defines which items to return.

Format: ATTRIB;LIST;item;
Status: ATTRIB;STATUS;statuscode;statusmessage;
Result: ATTRIB;RESULT;0;item data1;
Result: ATTRIB;RESULT;0;item data2;
Result: ATTRIB;RESULT;1;item dataN;

Valid items are:

INPUT	
Item	**Description**
SAPIOUT	Known SAPI output device names
SAPIVOICE	Known SAPI voices

OUTPUT	
Item	**Description**
item data1	Name of 1st output device or voice
item data2	Name of 2nd output device or voice
item dataN	Name of Nth output device or voice

Chapter Five
CMSG Operations

This chapter describes the ***First*PAGE** protocol subset known as the CMSG operations. These commands allow the developer to manipulate common message lists.

LIST

The LIST sub-operation facilitates the interrogation of the known common messages for the currently logged in account.

Format: CMSG;LIST;
Status: CMSG;STATUS;statuscode;statusmessage;
Result: CMSG;RESULT;0;msgid1|message1;
Result: CMSG;RESULT;0;msgid2|message2;
Result: CMSG;RESULT;1;msgidN|messageN;

OUTPUT	
Item	**Description**
msgid1	
message1	
msgid2	
message2	
msgidN	
messageN	

REMOVE

The REMOVE sub-operation facilitates the removal of a personal common messages within the *First*PAGE Server for the currently logged in user account.

Format: CMSG;REMOVE;msgid;
Status: CMSG;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
msgid	

WRITE

The WRITE sub-operation facilitates the creation of a new personal common messages within the *First*PAGE Server for the currently logged in user account. Use –1 for new messages, existing msgid for updates.

Format: CMSG;WRITE;msgid;message;
Status: CMSG;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
msgid	
message	

Chapter Six
DEST Operations

This chapter describes the ***First*PAGE** protocol subset known as the DEST operations. These commands allow the developer to request alarms be initiated or terminated within the engine.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific destination's attributes. If the engine is able to locate the specified destination then its details are returned, if it is unable to locate the destination an error will be returned.

Format: DEST;DETAIL;destname;
Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;eof;dest|title|email|pid|nid|tzid|type|note|id;

INPUT	
Item	**Description**
destname	Destination name to lookup

OUTPUT	
Item	**Description**
dest	Destination name
title	Destination title
email	Destination email
pid	Provider id
nid	Network Id
tzid	Time zone Id (See DEST;TZONES;)
type	Destination type (See DEST;TYPES;)
note	Note on the destination
id	Destination id

INITIATE

The INITIATE sub-operation facilitates the initiation of messaging transactions within the engine. If the engine is able to accept the specified transaction then a serial number for this transaction is returned. This serial number may be used to monitor or cancel the request.

Format: DEST;INITIATE;destname;message;
Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;1;serialnumber;

INPUT	
Item	**Description**
destname	Destination name
message	Message text (Note ; and \ must be escaped)

OUTPUT	
Item	**Description**
serialnumber	Assigned serial number

Format: DEST;INITIATE;@distlistname;message;
Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;1;serialnumber;

INPUT	
Item	**Description**
@distlistname	Distribution list name
message	Message text (Note ; and \ must be escaped)

OUTPUT	
Item	**Description**
serialnumber	Assigned serial number

LIST

The LIST sub-operation facilitates the interrogation of the known destinations that exist within the *First*PAGE Server engine.

Format: DEST;LIST;

Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;0;dest|title;type;netid;a1;a2,a3;loc;
Result: DEST;RESULT;1;…

OUTPUT	
Item	**Description**
dest	Name
title	Title
type	Destination type (See DEST;TYPES;)
netid	Network ID
a1	Alias 1
a2	Alias 2
a3	Alias 3
loc	Locator

QUEUE

The QUEUE sub-operation facilitates the direct injection of transactions into a providers queue. If it is unable to locate the provider/coverage, an error will be returned. If the engine is able to accept the specified transaction then a serial number for this transaction is returned. This serial number may be used to monitor or cancel the request.

Format: DEST;QUEUE;provider;coverage;networkid;msg;
Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;1;serialnumber;;

INPUT	
Item	**Description**
provider	Provider name
coverage	Coverage name
networkid	Network Id
msg	Message text (Note ; and \ must be escaped)

OUTPUT	
Item	**Description**
serialnumber	Assigned serial number

RELOAD

The RELOAD sub-operation indicates to the *First*PAGE Server that it should reload all internal tables related to destinations.

Format: DEST;RELOAD;
Status: DEST;STATUS;statuscode;statusmessage;

TYPES

The TYPES sub-operation facilitates the interrogation of the known destination types that exist within the *First*PAGE Server engine.

Format: DEST;TYPES;
Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;0;type|name|display|;
Result: DEST;RESULT;1;…

OUTPUT	
Item	**Description**
type#	Type Number (eg. 1)
name	Type Name (eg. ALPHAPAGER)
display	Display Name (eg. Alpha Pager)

TZONES

The TZONES sub-operation facilitates the interrogation of the known destination time zones that exist within the *First*PAGE Server engine.

Format: DEST;TZONES;
Status: DEST;STATUS;statuscode;statusmessage;
Result: DEST;RESULT;0;zoneid|name|display|;
Result: DEST;RESULT;1;…

OUTPUT	
Item	**Description**
zoneid	Zone id #
name	Name
display	Display name

Chapter Seven
DIST Operations

This chapter describes the ***First*PAGE** protocol subset known as the DIST operations. These commands allow the developer to manipulate transactions involving distribution lists.

INITIATE

The INITIATE sub-operation facilitates the initiation of distribution list transactions within the engine. If the engine is able to accept the specified transaction then a serial number for this transaction is returned. This serial number may be used to monitor or cancel the request.

Format: DIST;INITIATE;@distlist;message;
Status: DIST;STATUS;statuscode;statusmessage;
Result: DIST;RESULT;1;serialnumber;

INPUT	
Item	**Description**
@distlistname	Distribution list name
Message	Message text (Note ; and \ must be escaped)

OUTPUT	
Item	**Description**
Serialnumber	Assigned serial number

NOTE: Distribution list initiates utilize the DEST function rather than the DIST function.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific distribution list's members. If the engine is able to locate the specified distribution list, then its details are returned. If it is unable to locate the distribution list, an error will be returned.

Format: DIST;DETAIL;name;
Status: DIST;STATUS;statuscode;statusmessage;
Result: DIST;RESULT;0;dest|title|email|#|#|#|#|note|#;
Result: DIST;RESULT;1;…

INPUT	
Item	**Description**
name	Name of destination to lookup

LIST

The LIST sub-operation facilitates the interrogation of the known distribution lists.

Format: DIST;LIST;
Status: DIST;STATUS;statuscode;statusmessage;
Result: DIST;RESULT;0;dist;description;
Result: DIST;RESULT;1;…

OUTPUT	
Item	**Description**
dist	Distribution list name
description	Description

RELOAD

The RELOAD sub-operation indicates to the *First*PAGE Server that it should reload all internal tables related to distribution lists.

Format: DIST;RELOAD;
Status: DIST;STATUS;statuscode;statusmessage;

Chapter Eight
GROUP Operations

This chapter describes the ***First*PAGE** protocol subset known as the GROUP operations. These commands allow the developer to request information on destination groups defined within the *First*PAGE Server.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific group's members. If the engine is able to locate the specified group, then its details are returned. If it is unable to locate the group, an error will be returned.

Format: GROUP;DETAIL;name;
Status: GROUP;STATUS;statuscode;statusmessage;
Result: GROUP;RESULT;0;DestID;
Result: GROUP;RESULT;0;DestID2;
Result: GROUP;RESULT;1;DestIDN;

INPUT	
Item	**Description**
name	Group name

OUTPUT	
Item	**Description**
DestID	
DestID2	
DestIDN	

LIST

The LIST sub-operation facilitates the interrogation of the known groups.

Format: GROUP;LIST;
Status: GROUP;STATUS;statuscode;statusmessage;

Result: GROUP;RESULT;0;group;descr;
Result: GROUP;RESULT;1;…

OUTPUT	
Item	**Description**
group	Group name
descr	Description

RELOAD

The RELOAD sub-operation indicates to the *First*PAGE Server that it should reload all internal tables related to groups.

Format: GROUP;RELOAD;
Status: GROUP;STATUS;statuscode;statusmessage;

Chapter Nine
PDEST Operations

This chapter describes the ***First*PAGE** protocol subset known as the PDEST operations. These commands allow the developer to initiate and manage personal destination transactions. Personal destinations are user specific destinations that users may maintain for providers defined within the *First*PAGE Server configuration.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific personal destination's attributes. If the engine is able to locate the specified personal destination then its details are returned. If it is unable to locate the personal destination, an error will be returned.

Format: PDEST;DETAIL;destname;
Status: PDEST;STATUS;statuscode;statusmessage;
Result:
PDEST;RESULT;1;name|title|email|provid|netid|tzid|typeid|notes;

INPUT	
Item	**Description**
destname	Personal destination to lookup

OUTPUT	
Item	**Description**
name	Personal destination name
title	Title
email	Email address
provid	Provider Id
netid	Network Id
tzid	Time Zone Id (See DEST;TZONES;)
typeid	Type Id (See DEST;TYPES;)
notes	Note entered

INITIATE

The INITIATE sub-operation facilitates the initiation of messaging transactions within the engine. If the engine is able to accept the specified transaction then a serial number for this transaction is returned. This serial number may be used to monitor or cancel the request.

Format: PDEST;INITIATE;destname;message;
Status: PDEST;STATUS;statuscode;statusmessage;
Result: PDEST;RESULT;1;serialnumber;

INPUT	
Item	**Description**
destname	Personal destination name
message	Message text (Note ; and \ must be escaped)

OUTPUT	
Item	**Description**
serialnumber	Serial number assigned to transaction

LIST

The LIST sub-operation facilitates the interrogation of the known destinations that exist within the *First*PAGE Server engine for the currently logged in user.

Format: PDEST;LIST;
Status: PDEST;STATUS;statuscode;statusmessage;
Result: PDEST;RESULT;0;destination1;
Result: PDEST;RESULT;0;destination2;
Result: PDEST;RESULT;1;destinationN;

OUTPUT	
Item	**Description**
destination1	1ST personal destination name
destination2	2nd personal destination name
destinationN	Nth personal destination name

REMOVE

The REMOVE sub-operation facilitates the removal of a personal destination within the *First*PAGE Server for the currently logged in user account.

Format: PDEST;REMOVE;name;
Status: PDEST;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
name	Personal destination name to remove

WRITE

The WRITE sub-operation facilitates the creation of a new. Or update of an existing personal destination within the *First*PAGE Server for the currently logged in user account.

Format: PDEST;WRITE;name|title|email|provid|netid|tzid|typeid|destinationtypeid;notes;
Status: PDEST;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
name	
title	
email	
provid	
netid	
tzid	
typeid	
destinationtypeid	
notes	

Chapter Ten
PDIST Operations

This chapter describes the ***First*PAGE** protocol subset known as the PDIST operations. These commands allow the developer to initiate and manage personal destination distribution list transactions. Personal destination distribution lists are user specific distribution lists that users may maintain for providers defined within the *First*PAGE Server configuration.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific personal distribution's attributes. If the engine is able to locate the specified personal distribution list, then its details are returned. If it is unable to locate the personal distribution list, an error will be returned.

Format: PDIST;DETAIL;distname;
Status: PDIST;STATUS;statuscode;statusmessage;
Result: PDIST;RESULT;0;DestID;
Result: PDIST;RESULT;0;DestID2;
Result: PDIST;RESULT;1;DestIDN;

INPUT	
Item	**Description**
distname	Personal distribution list name to retrieve details for

OUTPUT	
Item	**Description**
DestID	
DestID2	
DestIDN	

INITIATE

The INITIATE sub-operation facilitates the initiation of messaging transactions within the engine. If the engine is able to accept the specified transaction then a serial number for this transaction is returned. This serial number may be used to monitor or cancel the request.

Format: PDIST;INITIATE;distname;message;
Status: PDIST;STATUS;statuscode;statusmessage;
Result: PDIST;RESULT;1;serialnumber;

INPUT	
Item	**Description**
distname	Personal distribution list name to send message too
message	Message to send to distribution list

OUTPUT	
Item	**Description**
serialnumber	Serial number for transaction

LIST

The LIST sub-operation facilitates the interrogation of the known personal distribution lists for the currently logged in account.

Format: PDIST;LIST;
Status: PDIST;STATUS;statuscode;statusmessage;
Result: PDIST;RESULT;0;distlist1;
Result: PDIST;RESULT;0;distlist2;
Result: PDIST;RESULT;1;distlistN;

OUTPUT	
Item	**Description**
distlist1	1st personal distribution list name
distlist2	2nd personal distribution list name
distlistN	Nth personal distribution list name

REMOVE

The REMOVE sub-operation facilitates the removal of a personal distribution list within the *First*PAGE Server for the currently logged in user account.

Format: PDIST;REMOVE;name;
Status: PDIST;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
name	Personal distribution list name to remove

WRITE

The WRITE sub-operation facilitates the creation of a new personal distribution list within the *First*PAGE Server for the currently logged in user account.

Format: PDIST;WRITE;name;notes;pdest1|pdest2|pdestN|;
Status: PDIST;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**

name	
notes	
pdest1	
pdest2	
pdestN	

Chapter Eleven
PROV Operations

This chapter describes the ***First*PAGE** protocol subset known as the PROV operations. These commands allow the developer to request information on the service providers define in the *First*PAGE Server.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific provider. If the engine is able to locate the provider, then its details are returned. If it is unable to locate the provider, an error will be returned.

Format: PROV;DETAIL;provider;coverage;
Status: PROV;STATUS;statuscode;statusmessage;
Result: PROV;RESULT;1;name|coverage|address|speed|maxsec|maxtrans|maxalpha|maxnum|protocol|bytesize|stopbits|parity|comp|errorcorr;

INPUT	
Item	**Description**
provider	Provider name to lookup
coverage	Coverage name to lookup

OUTPUT	
Item	**Description**
name	Name of provider
coverage	Coverage name of provider
address	
speed	
maxsec	
maxtrans	
maxalpha	Maximum alpha numeric length
maxnum	Maximum numeric length
protocol	Protocol name

bytesize	Byte size if serially connected
stopbits	Stop bits if serially connected
parity	Parity if serially connected
comp	Compression YES or NO if modem connection
errorcorr	Error Correction YES or NO if modem connection

FAILBACK

FAILBACK sub-operation facilitates releasing a previous manual failover invocation of the specified provider.

Format: PROV;FAILBACK;provider;coverage;
Status: PROV;STATUS;statuscode;statusmessage;

OUTPUT	
Item	**Description**
provider	Provider name
coverage	Coverage name

FAILOVER

FAILOVER sub-operation facilitates the manual failover invocation of the specified provider.

Format: PROV;FAILOVER;provider;coverage;
Status: PROV;STATUS;statuscode;statusmessage;

OUTPUT	
Item	**Description**
provider	Provider name
coverage	Coverage name

LIST

LIST sub-operation facilitates the interrogation of the known providers.

Format: PROV;LIST;
Status: PROV;STATUS;statuscode;statusmessage;
Result: PROV;RESULT;0;name1|coverage1;
Result: PROV;RESULT;0;name2|coverage2;
Result: PROV;RESULT;1;nameN|coverageN;

OUTPUT	
Item	**Description**
name#	Provider name
coverage#	Coverage name

RELOAD

The RELOAD sub-operation indicates to the *First*PAGE Server that it should reload all internal tables related to providers.

Format: PROV;RELOAD;
Status: PROV;STATUS;statuscode;statusmessage;

STATE

STATE sub-operation facilitates querying the failover state of the specified provider.

Format: PROV;STATE;provider;coverage;
Status: PROV;STATUS;statuscode;statusmessage;
Result: PROV;RESULT;1;FAILOVERSTATE|failoverstate;

OUTPUT	
Item	**Description**
provider	Provider name
coverage	Coverage name

failoverstate	0 – normal, 1 – automatic, 2 – manual

TRANSCOUNT

TRANSCOUNT sub-operation facilitates querying the number of transactions in the queue of the specified provider.

Format: PROV;TRANSCOUNT;provider;coverage;
Status: PROV;STATUS;statuscode;statusmessage;
Result: PROV;RESULT;1;QUEUESIZE|transcount|sizekB;

OUTPUT	
Item	**Description**
provider	Provider name
coverage	Coverage name
transcount	Number of transactions in queue
sizekB	Size of queue in kilobytes

Chapter Twelve
RECUR Operations

This chapter describes the ***First*PAGE** protocol subset known as the RECUR operations. These commands allow the developer to submit, modify and remove recurring message transactions from the engine.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific recurring transaction's attributes. If the engine is able to locate the specified transaction then its details are returned, if it is unable to locate the transaction an error will be returned.

Format: RECUR;DETAIL;serialnumber;
Status: RECUR;STATUS;statuscode;statusmessage;
Result:
RECUR;RESULT;1;id|username|domain|destination|message
|sendtime|type|detail|startdate|rangetype|nextsenddate|prevs
enddate|range#or-date;

INPUT	
Item	**Description**
serialnumber	

OUTPUT	
Item	**Description**
id	
username	
domain	
destination	
message	
sendtime	
type	
detail	
startdate	
rangetype	

nextsenddate	
prevsenddate	
range#or-date	

LIST

The LIST sub-operation facilitates the interrogation of the known recurring transactions within the *First*PAGE Server for the currently logged in user account.

Format: RECUR;LIST;
Status: RECUR;STATUS;statuscode;statusmessage;
Result: RECUR;RESULT;0;recurid1;
Result: RECUR;RESULT;0;recurid2;
Result: RECUR;RESULT;1;recuridN;

OUTPUT	
Item	**Description**
recurid1	
recurid2	
recuridN	

REMOVE

The REMOVE sub-operation facilitates the removal of a recurring transaction within the *First*PAGE Server for the currently logged in user account.

Format: RECUR;REMOVE;serialnumber;
Status: RECUR;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
serialnumber	

WRITE

The WRITE sub-operation facilitates the initiation of a new recurring transaction within the *First*PAGE Server for the currently logged in user account.

Format: RECUR;WRITE;recurid;dest;msg; time;
type;details;start;rangetype;<optional rangevalue>;
Status: RECUR;STATUS;statuscode;statusmessage;
Result: RECUR;RESULT;1;recurid;

INPUT	
Item	**Description**
recurid	Serial number of recurrence to update, -1 to create new
dest	Destination name
msg	Message text
Time	hh:mm of when recurrence sends (24 hour format)
type	1 – Daily 2 – Weekly 3 – Monthly 4 – Yearly
details	<OPT-#> - Option number <EVERY-x> - Every x ???? <DOW-xxx> - Day of week (ie. SUN, FRI)

	<MONTH-#> - Month number, Jan = 1 <DAY-#> - Day of the month

Options		
Type	**Item**	**Description**
Daily	<OPT-1>	Every weekday
Daily	<OPT-2>	Every x days
Weekly	<OPT-1>	Every one of the specified DOW
Monthly	<OPT-1>	Day x of every y months
Monthly	<OPT-2>	xth DOW of every y months
Yearly	<OPT-1>	Every day x of month y
Yearly	<OPT-2>	xth DOW of MONTH

Chapter Thirteen
SESSION Operations

This chapter describes the ***First*PAGE** protocol subset known as the SESSION operations. These commands allow the developer to set session options.

WRITE

The WRITE sub-operation facilitates the setting of various values for the user session. Item may be HONORSCHED and its value may be ON or OFF or RETRIES and a count of >= 1 and <= 10. If HONORSCHED is set to ON then any schedule for a destination will be honoured. If OFF the schedule will be ignored and the destination will be sent any message regardless of the time of day. RETRIES value is an integer representing the number of times the message will be attempted to be delivered to the provider.

Format: SESSION;WRITE;item;value;
Status: SESSION;STATUS;statuscode;statusmessage;

INPUT	
Item	**Description**
Item	HONORSCHED or RETRIES
Value	HONORSCHED: ON or OFF RETRIES: Integer value >= 1, <= 10

Chapter Fourteen
TRANS Operations

This chapter describes the ***First*PAGE** protocol subset known as the TRANS operations. These commands allow the developer to request transaction information from the engine regarding transactions previously initiated.

DETAIL

The DETAIL sub-operation facilitates the querying of a specific transaction's attributes. If the engine is able to locate the specified transaction then its details are returned, if it is unable to locate the transaction an error will be returned.

Format: TRANS;DETAIL;serialnumber;
Status: TRANS;STATUS;statuscode|statusmessage|updatetime;
Result: TRANS;RESULT;1;status;

INPUT	
Item	**Description**
serialnumber	Transaction number of DEST;INITIATE;

OUTPUT	
Item	**Description**
status	Most recent status code of transaction.
statusmessage	Most recent status message of transaction.
updatetime	Most recent status update date/time (GMT).

LIST

The LIST sub-operation facilitates the interrogation of the known transactions within the *First*PAGE Server for the currently logged in user account. All transactions for which a status is known are returned.

Format: TRANS;LIST;
Status: TRANS;STATUS;statuscode;statusmessage;
Result: TRANS;RESULT;0;serialnumber1;
Result: TRANS;RESULT;0;serialnumber2;
Result: TRANS;RESULT;1;serialnumberN;

OUTPUT	
Item	**Description**
serialnumber1	
serialnumber2	
serialnumberN	

Chapter Fifteen
USER Operations

This chapter describes the ***First*PAGE** protocol subset known as the USER operations. These commands allow the developer to define user options and obtain status.

DETAIL

The DETAIL sub-operation facilitates a client application to obtain the authenticated users preferences within the *First*PAGE Server engine. The items returned are listed in the table found under the WRITE sub-op within the USER operations.

If the engine is running in a secure mode then the user login for the user initiating this command must be an administrator if the optional domain and username parameters are specified.

Format: USER;DETAIL;[domain;username;]
Status: USER;STATUS;statuscode;statusmessage;
Result: USER;RESULT;0;item1|item1value;
Result: USER;RESULT;0;item2|item2value;
Result: USER;RESULT;1;itemN|itemNvalue;

INPUT	
Item	**Description**
Domain	Domain name (optional)
Username	User name (optional)

OUTPUT		
Item	**Value(s)**	**Notes**
DESTINATION	destinationname @distlistname	Destination name(s) associated with

		this user account. Not returned if there are no associated destinations/DLs. There may be many of these values.Prefixed with “@” if a distribution list.
HONORSCHED	ON, OFF	Status of destination schedule honouring
IDLEMINUTES	0-99	0 = GUI will never lock 1-99 = GUI will lock out after x minutes, requiring user to re-authenticate to continue.
LOGONDOMAIN	domainname	Domainname of the account.
SECURITYLEVEL	Security level value	
SIGENVVAR	Environment variable name.	If SIGSTATE = 1 then the contents of this client side environment variable will be used by *First*PAGE Client for the signature.

SIGMETHOD	0-2	0 = Signature provided by client 1 = Signature retrieved from environment variable specified in SIGENVVAR 2 = Signature is authenticated username.
SIGSTATE	0-1	0 = Signature is off 1 = Signature is on
SIGVALUE	Text	Signature to be used within *First*PAGE Client
USERNAME	Username	Username of the logged in account
GROUPLIST	Text	Comma-separated list of associated destination groups

DESTLIST

The DESTLIST sub-operation facilitates a client application to obtain the authenticated users destinations within the *First*PAGE Server engine.

If the engine is running in a secure mode then the user login for the user initiating this command must be an administrator if the optional domain and username parameters are specified.

Format: USER;DESTLIST;[domain;username;]
Status: USER;STATUS;statuscode;statusmessage;
Result: USER;RESULT;eof;destinationtype|destinationname;

OUTPUT	
Item	**Description**
Destinationtype	DestinationType (1=destination, 2=distribution list)
Destinationname	Username of the user account

LIST

The LIST sub-operation facilitates the listing of the known users that exist within the FirstPAGE Server engine. If the engine is running in a secure mode then the user login for the user initiating this command must be an administrator.

Format: USER;LIST;
Status: USER;STATUS;statuscode;statusmessage;
Result: USER;RESULT;eof;domainname|username|usertype;

OUTPUT	
Item	**Description**
domainname	Domain of the user account
username	Username of the user account
usertype	User type (ADMIN or USER)

LOGIN

The LOGIN sub-operation facilitates the authentication or identification of the user to the *First*PAGE Server engine. If the engine is running in secure mode then the username and password must exist in the user table before the user may interact with *First*PAGE Server, otherwise a new user record is created using default preferences. If SECMODE is off then this record is only used to track user-specific preferences and not for authentication. In the instance of SECMODE authentication is Off then the login of: Domain=NCTI, User=ADMIN, Pass=ADMIN should be utilized.

Format: USER;LOGIN;domain;user;password;
Status: USER;STATUS;statuscode;statusmessage;

INPUT	
Field	**Description**
domain	Domain of user account. Note: If security mode is Ford Authentication then domain must be "FORD".
user	User account
password	Password

PASSCHG

The PASSCHG sub-operation facilitates a client application changing the password of an authenticated user within the *First*PAGE Server engine.

Format: USER;PASSCHG;oldpass;newpass;
Status: USER;STATUS;statuscode;statusmessage;

INPUT	
Field	**Description**
oldpass	
newpass	

TYPE

The TYPE sub-operation facilitates a client application determining the type of user currently logged in.

Format: USER;TYPE;
Status: USER;STATUS;statuscode;statusmessage;
Result: USER;RESULT;eof;usertype;

OUTPUT	
Field	**Description**
usertype	One of USER or ADMIN

WRITE

The WRITE sub-operation facilitates the setting of user preferences within the *First*PAGE Server engine. At this time the following values are supported:

Item	**Value(s)**	**Notes**
DESTINATION	destinationname	Destination name associated with this user account. Blank if no associated destination. (Note: Should not be listed as available for USER;WRITE; Once moved to USER;DETAIL; then remove it here)
HONORSCHED	ON, OFF	Determine if destination schedules should be

		honoured
SIGVALUE	Text	Signature to be used within *First*PAGE Client
SIGMETHOD	0-2	0 = Signature provided by client 1 = Signature retrieved from environment variable specified in SIGENVVAR 2 = Signature is authenticated username.
SIGSTATE	0-1	0 = Signature is off 1 = Signature is on
SIGENVVAR	Environment variable name.	If SIGSTATE = 1 then the contents of this client side environment variable will be used by *First*PAGE Client for the signature.
IDLEMINUTES	0-99	0 = GUI will never lock

		1-99 = GUI will lock out after x minutes, requiring user to re-authenticate to continue.

Format: USER;WRITE;item;value;
Status: USER;STATUS;statuscode;statusmessage;

INPUT	
Field	**Description**
item	
value	

Technical Assistance

FAQ

Every SeQent product contains a FAQ file specific to that product. You can view this FAQ file by visiting the SeQent web site (www.SeQent.com) and browse the product page for the product you are using.

Support Contracts

You may purchase a support contract for *First*PAGE based products. Various levels of support are available. View the SeQent web site (**www.SeQent.com**) and browse the product page for product you are using.

Help Desk

To reach our help desk please use one of the following methods:

Telephone: +1.519.652.0401
Fax: +1.519.652.9275
Web: www. SeQent.com - Customer Care
E-Mail: support@SeQent.com

Please have your support contract, or product license key ready, before calling or include it in your correspondence. Support is free for product trials and for the first 30 days of product ownership.

Glossary

Alias

A *First*PAGE Server alias is an alternate name, which may be assigned to a destination. *First*PAGE Server client applications may choose to allow users to utilize aliases to lookup actual destination names.

Client

A client component utilizes the facilities of a standard NETCON server. Typically a GUI application, this component provides a user-friendly interface of the available services offered by the server engine. An example of a client tool would be *First*PAGE Client, which presents the user with a list of known messaging destinations and allows them to send messages to one or more destinations.

Coverage

The coverage property of a provider is a name assigned to reflect the geographical area that this connection to the service provider allows you to send message too. Usually for example in the case of dial-up modem connection to service providers a different phone number is provider for each city. The coverage field provides the administrator an easy method of tracking all of the different coverage areas a single provider is supporting.

Destination

A *First*PAGE Server destination is a name assigned to a messaging location to which text messages may be delivered. Ex. SCOTTS_PAGER, KENS_PHONE, JIMS_EMAIL.

Device

A *First*PAGE Server device is a name assigned to a physical connection point on the server that will be used to establish a connection with a service provider.

Distribution List

A *First*PAGE Server distribution list is a collection of destinations that can be sent a message with a single transaction. Similar to an e-mail distribution list a client that initiates a transaction with a distribution list will automatically send the same message to all associated destinations.

Engine
An engine component implements a unified interface to a set of common, related protocols. The engine provides services to clients and gateways such that a common interface may be used to manipulate many different types of devices. Engines are administered via Administrator components and are utilized by clients, gateways or use written components.

Gateway
A gateway component creates a bridge between a third party product and a standard NETCON server. Typically an engine this component provides a conduit for forwarding events from the third party product into a NETCON standard server. An example of a gateway would be an interface to an HMI or SCADA system forwarding events from that system to *First*PAGE Alarm Manager for processing. An example of a gateway is Marquee Manager Gateway for VisualPlant. This gateway forwards VisualPlant incidents to Marquee Manager.

Group
A *First*PAGE Server group is a name assigned to a collection of destinations that are logically connected. Groups are used by *First*PAGE Server clients to limit the destinations which are available for use.

Network ID
A destination network id is the identifier the messaging network requires to remit a message to an end destination. Similar to a phone number a network id can be any NETCON Technologies number of numbers, alphabetic characters, or a combination of the two. For TAP connected providers the network id is usually a numeric value. For SMTP connected providers the network id is the devices assigned e-mail address.

Numeric Length
The Numeric Length property of a provider provides the same functionality as the Alpha Length property.

Provider
A *First*PAGE Server provider is a name assigned to the configuration information required to connect to a messaging service provider.

Schedule
A *First*PAGE Server schedule is a name assigned to a set of weekday configuration rules, which define the times during which a destination may receive messages. A schedule, once defined may then be assigned to a destination over a time period.

Example Sessions

Before attempting to use any of these interfaces please be sure to read the Packet Format section of Chapter 2.

Connect & Login

Confirm FirstPAGE Server Connection

- Performed one time when first connecting the socket
- Ensure the 4th parameter in the 2nd response is “FirstPAGE Server”

```
ADMIN;DETAIL;
ADMIN;STATUS;0;Success;
ADMIN;RESULT;1;FirstPAGE Server|VM2W2K3SE2|9|v9.08.10|IX86-Win32|2013-
06-20 18:3
1:00;
```

Login to Generic Account:

- Performed one time when first connecting the socket
- Ensure the status value returned is “0”

```
USER;LOGIN;NCTI;ADMIN;ADMIN;
USER;STATUS;0;Success;
```

Sending Message to a Destination/DL

Perform Connect & Login

Initiate a message to Jim’s Phone:

- Performed once for each destination to be notified.
- Ensure the status value is “0” before looking for response with transaction serial number.

```
DEST;INITIATE;JIMSPHONE;Jim lets meet for lunch at 13:00 @ Franks;
DEST;STATUS;0;Success;
DEST;RESULT;1;VM220035800240065;
```

Initiate a message to the distribution list for telecom team:

- Performed once for each destination to be notified.
- Ensure the status value is “0” before looking for response with transaction serial number.

```
DEST;INITIATE;@TELECOM;Are are all going to meet for lunch at 13:00 @ Franks;
DEST;STATUS;0;Success;
DEST;RESULT;1;VM220035800242341;
```

List of Known Destinations

Perform Connect & Login

Request List of Known Destinations:

- Performed once to obtain list of all known destinations. RESULT;1; when list is complete.
- Ensure the status value is “0” before looking for response with list of destination names.

```
DEST;LIST;
DEST;STATUS;0;Success;
DEST;RESULT;0;BURNS,SCOTT|CTO;1;SCOTTSPHONE@SEQENT.COM;;;;;
DEST;RESULT;0;JIMSPHONE|CFO;1;JIMSPHONE@SEQENT.COM;;;;;
DEST;RESULT;0;MACLENNAN,KATHY|Docs;1;1234567890;;;;;
DEST;RESULT;1;SAWKAR,SANDESH|ProjMgmt;3;2482491111;;;;;
```

List of Known Distribution Lists

Perform Connect & Login

Request List of Known Distribution Lists:

- Performed once to obtain list of all known distribution lists. RESULT;1; when list is complete.
- Ensure the status value is “0” before looking for response with list of distribution list names.

```
DIST;LIST;
DIST;STATUS;0;Success;
DIST;RESULT;0;PAINTSHOP|Paint shop staff;
```

DIST;RESULT;0;ITSUPPORT|IT Support Team;
DIST;RESULT;1;FINANCE|Finance Members;

Sending Message to a Unnamed Destination Network ID

Perform Connect & Login

Initiate a message to 5195551212:

- Performed once for each destination to be notified.
- Ensure the status value is “0” before looking for response with transaction serial number.
- In this example the named provider in FirstPAGE Server is BELLMOB and coverage is CANADA

DEST;QUEUE;BELLMOB;CANADA;5195551212@TXT.BELLMOBILITY.CA;Jim lets meet for lunch at 13:00 @ Franks;
DEST;STATUS;0;Success;
DEST;RESULT;1;VM220035800240067;