Connect:Express

Sterling Integrator Services

Version 5.0.05

Sterling Commerce
An IBM Company

Connect:Express
Sterling Integrator Services

Version 5.0.05
First Edition

Contents

Preface

This document describes the Connect:Express Services proposed in Sterling Integrator (SI). Connect:Express services enable you to use Connect:Express functionality from SI.

Overview describes the installation procedure and provides an overview of the Connect:Express services.

Reference Guide is the reference guide of Connect:Express services.

Notifications describes the implementation of Connect:Express notifications with SI.

Connect:Express User Interface describes the implementation of a Connect:Express browser interface with SI.

Introduction

Connect:Express SI services enable a SI business process to use the functionality of a Connect:Express Unix or Windows, locally or remotely.

The services provide an interface to the various components of Connect:Express:

- Symbolic partners and files: list, create, update, delete.
- Session and presentation tables: list and update.
- Connect:Express configuration file: list
- File transfers: active, journal and statistics.
- Submit a file transfer request.
- Interrupt, restart, purge a file transfer.

The services logon to Connect:Express via a TCP/IP connection.

Connect:Express services are available for Unix and Windows platforms.
Connect:Express Windows must be licensed for the « Activity Manager » option.

In the following, a Connect:Express monitor is referred to as a « monitor », the SI Browser interface (dashboard) is referred to as the « SI interface» and the Business process graphical modeler (GPM) is referred to as the « modeler ».

Reference Guide provides a detailed description of all services.

The browser interface enables trhe following operations:

- Symbolic partners and files: list, create, update, delete.
- File transfers: active, journal and statistics.
- Submit a file transfer request.
- Interrupt, restart, purge a file transfer.

This chapter provides an overview of the installation procedure and of the Connect:Express services.

Overview

Connect:Express SI services establish TCP/IP connections with monitors. You must configure an API TCP/IP port for each monitor that you want to administrate. You can configure the connection parameters in an instance of a Connect:Express service or pass them in the business process.
You can access all Connect:Express configuration files (Partners, Files, Tables) and manage file transfers.

You are provided a number of services and parameters. You can set default parameters in service instances that you configure and that you select in a business process. You can complement the parameters of the service in the business process itself.

Important note: the file transfer request service does not transfer any file from SI to Connect:Express. This means that you must place the file in the Connect:Express environment before sending the transfer request. For this purpose, you can use the SI File System Adapter, if the monitor is on the same system as SI, or FTP for example, if the monitor is remote.

You should take care of access rights to the files.

Supported Versions

The following table shows the correspondence between SI and Connect:Express versions supported:

Connect:Express SI Services	SI	Connect:Express Unix	Connect:Express Windows
5.0.05	5.0	143-1 with Patch 143-121	305.001

Installation

Connect:Express SI services are shipped in a zip file for Windows CXSI_WIN_5.0.05.zip or a tar file for Unix CXSI_UNIX_5.0.05.tar. When the installation is complete, Connect:Express services are shown in the Service Configuration screen of SI interface (Dashboard) and in the modeler (GPM).

Installing on Unix

Logon to the SI user account and do the following (we suppose that the SI installation directory is <installdir>):

1. Change directory to <installdir>/bin of SI, and stop SI if it is running.

   In a test environment:

```
# ./hardstop.sh
```

   In a production environment:

```
# ./softstop.sh
```

2. Copy CXSI_UNIX_5.0.05.tar in a temporary directory.
3. Extract the archive:

```
# tar xvf CXSI_UNIX_5.0.05.tar
```

4. The following files are extracted:

   - cx_5.0.05.jar
   - cxcmd.jar

- CXJAI.jar
- cxlogger.txt
- installer.jar
- install.sh

5. Install the services:

```
# ./install.sh
```

6. Restart SI.

```
# cd <installdir>/bin
# ./run.sh
```

Installing on Windows

Logon to the SI user account and do the following (we suppose that the SI installation directory is <installdir>):

1. Open a DOS command windows, change directory to <installdir>\bin of SI, and stop SI if it is running.

   In a test environment:

```
# hardstop.cmd
```

   In a production environment:

```
# softstop.cmd
```

2. Copy CXSI_WIN_5.0.05.zip in a temporary folder, for example c:\tmpdir.

3. Extract the archive (with WinZip for example).

4. If the present version of Sterling Integrator is an upgrade of GIS 4.3/4.2 and if the services of Connect:Express for GIS were already installed in the previous version, it is necessary to uninstall the previous version before installing this new version of the services (The products name has indeed been changed from Connect_Express_GIS_Adapters to Connect_Express_SI_Adapters). In that case, use the Add or Remove program utility of the Control Panel to uninstall.

5. **IMPORTANT: VERIFY THAT THE STERLING INTEGRATOR DATABASE SERVER IS STARTED BEFORE BEGINNING TO INSTALL THE SERVICES.** (If the database server of SI is installed as a Windows service, verify that this Windows service is started).

6. Start setup.exe to install the services.

7. Restart SI.

```
# run.cmd
```

Configuring the Listening Port of Connect:Express

You activate the listening port of Connect:Express differently on Windows and Unix.

Windows

To accept commands from SI services, Connect:Express Windows must have the "Activty Manager" option in its license key.

Connect:Express Windows monitor can listen to remote API clients. The default port number is 7000. You can display and update the active port number in the Graphical interface of Connect:Express through the dialog box « Administration \ Parameters \ Monitor \ Network \ TCP/IP ».

Unix

Connect:Express unix does not need any special license to accept commands from SI services.

You can configure the API listening port in the sysin file of Connect:Express in $TOM_DIR/config.

Edit the sysin file and add the line:

```
APPORT=<port-number>
```

You need to stop/start the monitor to activate the API server, tom_api process

Description of Connect:Express Services

The following table shows the list of the services provided:

Service	Description
CXTransferSubmit	Submit a transfer request to Connect:Express
CXTransferStatus	Inquire the status of a file transfer (Ended or running)
CXTransferControl	Interrupt, restart, purge a file transfer
CXPartner	Manage symbolic partners.
CXFile	Manage symbolic files.
CXSession	Manage PeSIT and Etebac3 session tables.
CXPresentation	Manage PeSIT and Etebac3 presentation tables of Connect:Express Unix Manage PeSIT presentation tables of Connect:Express Windows.
CXEtb3Presentation	Manage Etebac3 presentation tables of Connect:Express Windows.
CXJournal	List of the Journal.
CXActivity	List of active transfers.
CXConfiguration	List the configuration file and the asset protection information.
CXStatistics	List of the statistics.

A service enable you to access a component (Transfer request, Partner, File, Table …) with different modes. The following table shows the list of the modes available:

Mode	Description
CREATE	Creation. A new element is added, an error is returned if it is a duplicate.
REPLACE	Replace. A new element is added, it is replaced if it is a duplicate.
UPDATE	Update. The element is updated with the parameters provided.
GET	Read one or several elements. Characteristics of the elements selected are recorded in a document.
DELETE	Delete an element.
LIST	Read several elements. The lsit of elements selected is recorded in a document.
SUBMIT	Submit a file transfer request.
INTERRUPT	Interrupt a file transfer.
RESTART	Restart a file transfer.
PURGE	Purge one or several transfer requests

All modes are not available for each service: for example REPLACE is not available CXJournal service.

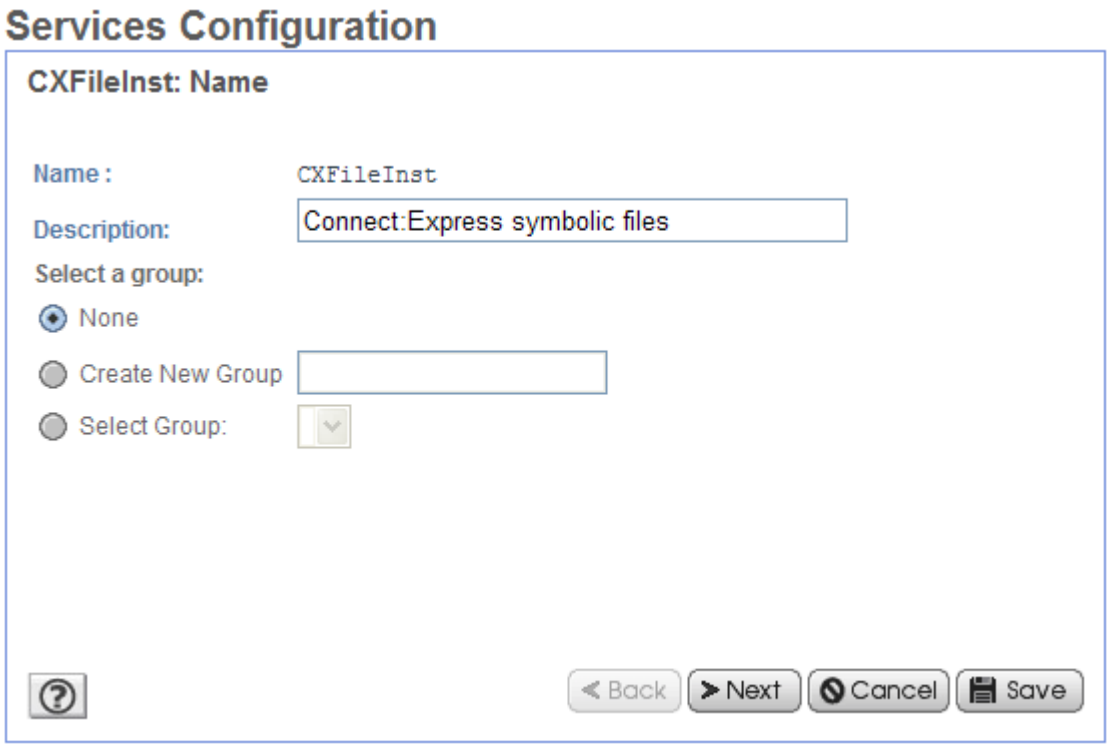
Implementing a Connect:Express Service

To implement a Connect:Express service do the following:

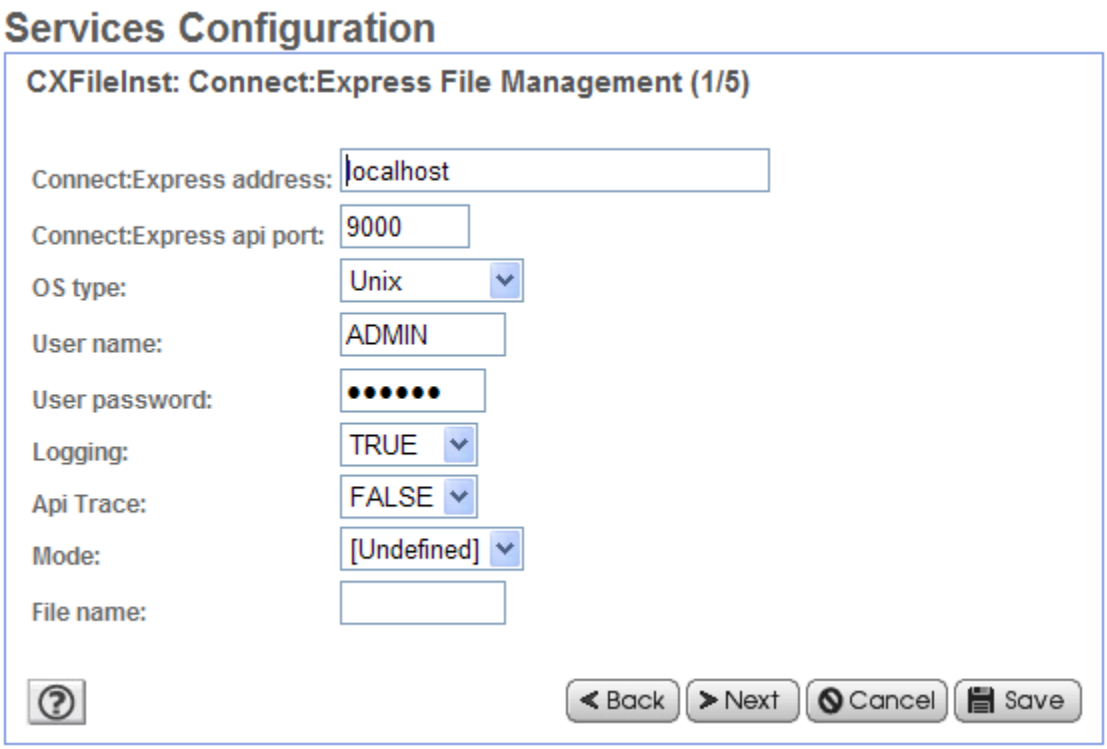
1. Create a service instance with SI interface.
2. Create a bpml source file with the modeler, to use this instance.
3. Import the business process source code with the SI interface: check-in the bpml file that you created.

Creating a Service Instance

Use Deployment/Services/Configuration/New Service of SI interface to create a new instance of a service. Connect:Express services are shown in the list, starting with "CX".



Fill in the parameters of the instance. You are not required to define all parameters: some can be left empty or set to [Undefined]. You will be able to provide them when creating the Business process. If you are creating an instance for one monitor, you can then fill in the IP address, listening port, OS type and Login information.



Some parameters are OS dependant. For example SNA LU6.2 is available for Connect:Express Windows only: in this case the name of the OS is indicated in the parameter label .
While executing the service, parameters that are not applicable are ignored.

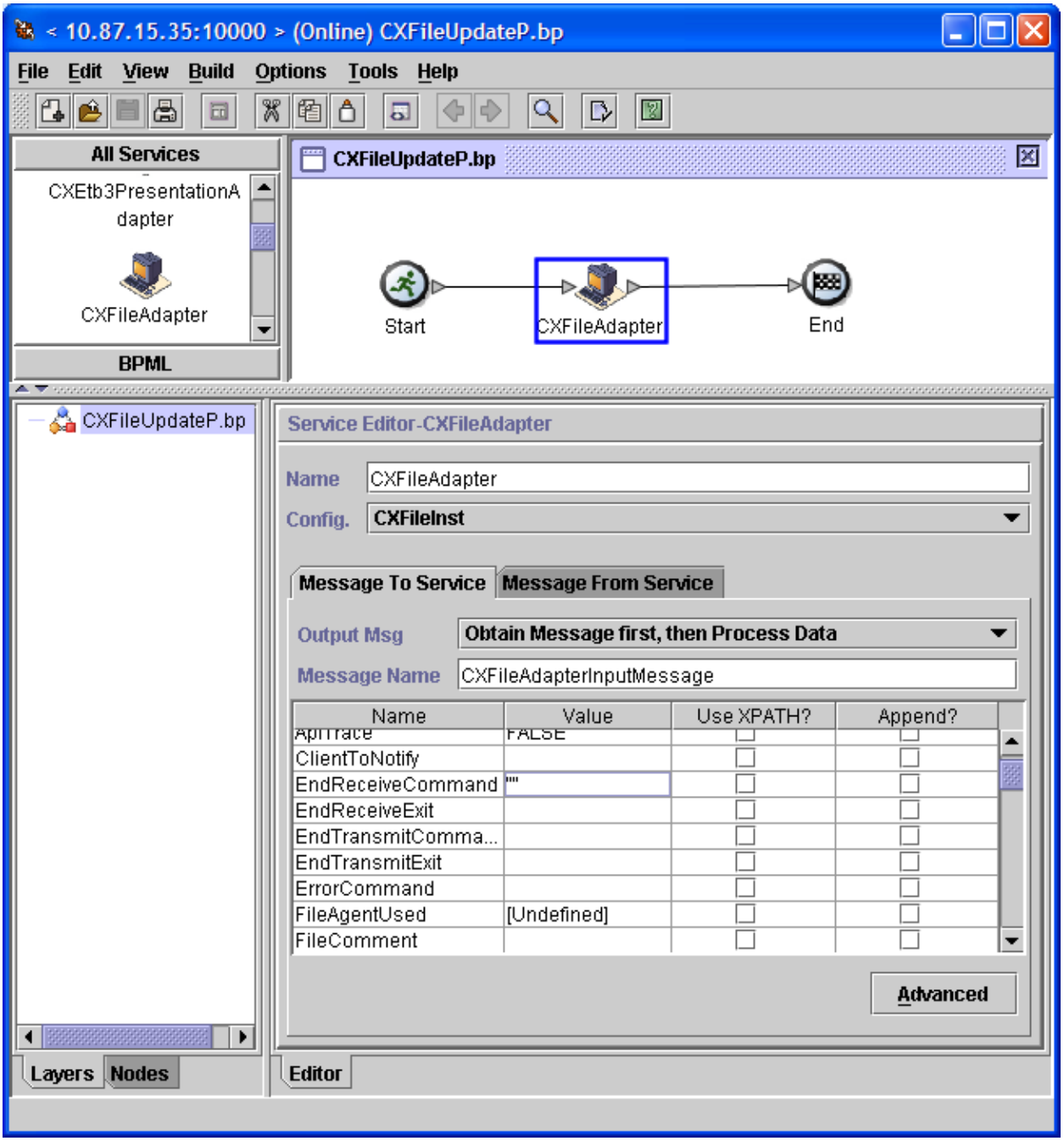
Creating a Business Process

The service instance that you created is available when configuring the service in a business process. The example below shows how the modeler screen displays a Connect:Express service instance and parameters.

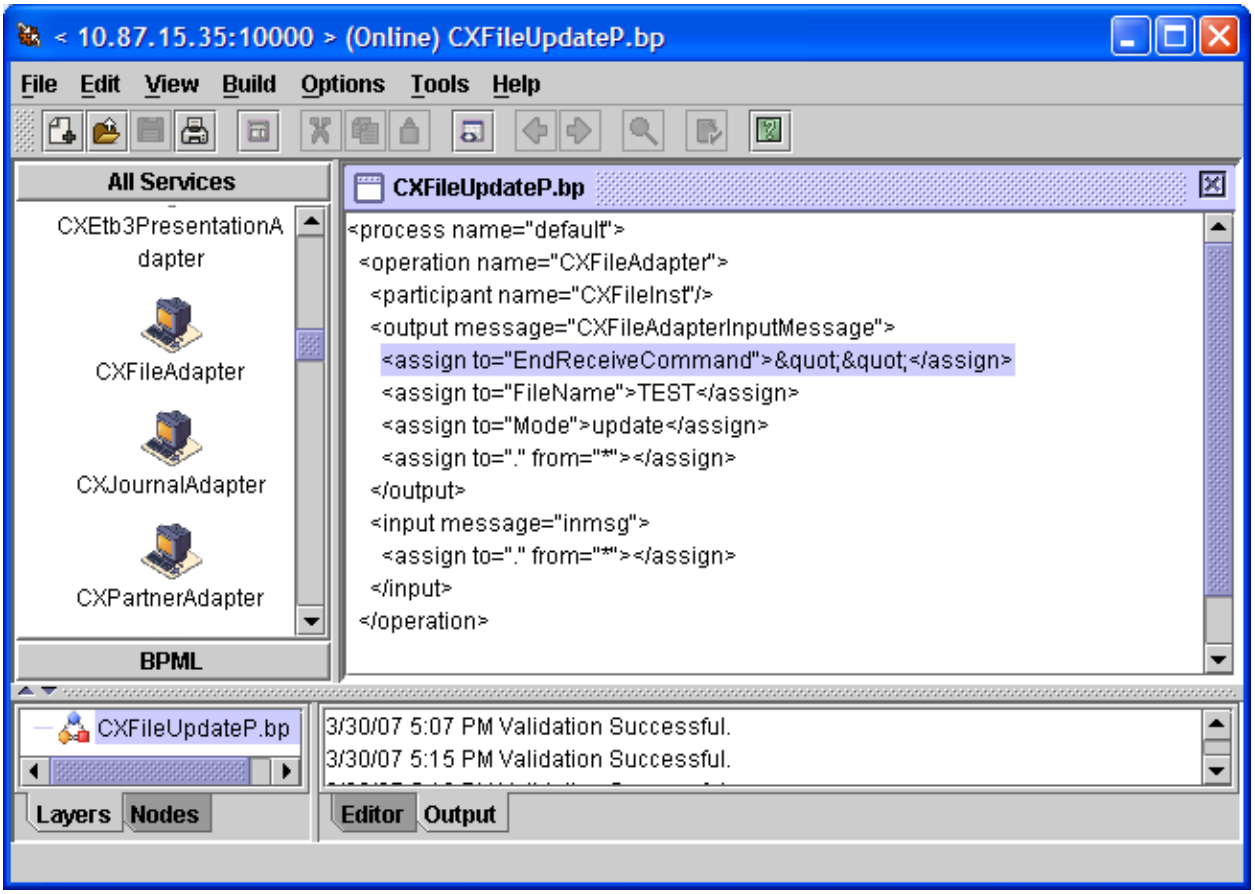
You can change the parameters that you defined in the instance. You can define the parameters that you did not define in the instance, using constants or XPATH expressions.

In order to clear a parameter, use «’‘’» (double quote, between quotes) as shown for the EndReceiveCommand field in the example. If you let the field blank, this means that you don’t define the parameter.

In UPDATE mode letting the field blank will not change the field, setting the field to « ‘’ » will set the field to spaces or zero according to the type of field.

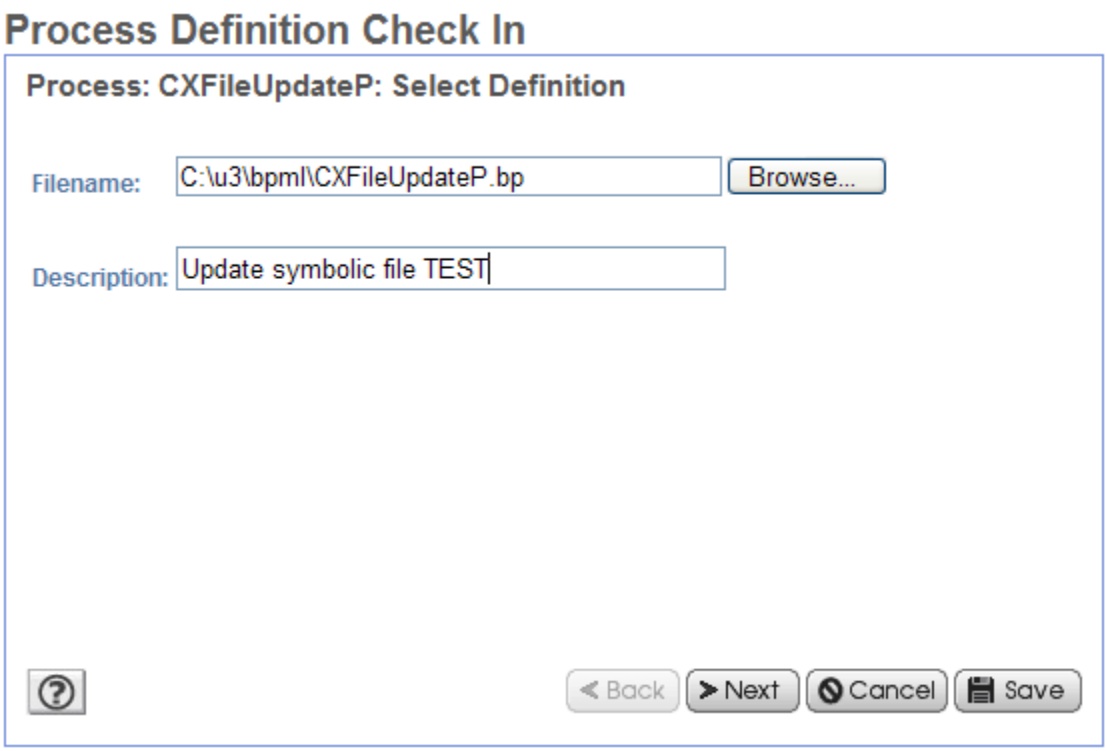


You can view the result in the bpml source code screen, as shown below:



Checking-in the Business Process

Use SI interface to check-in the Business Process in SI.



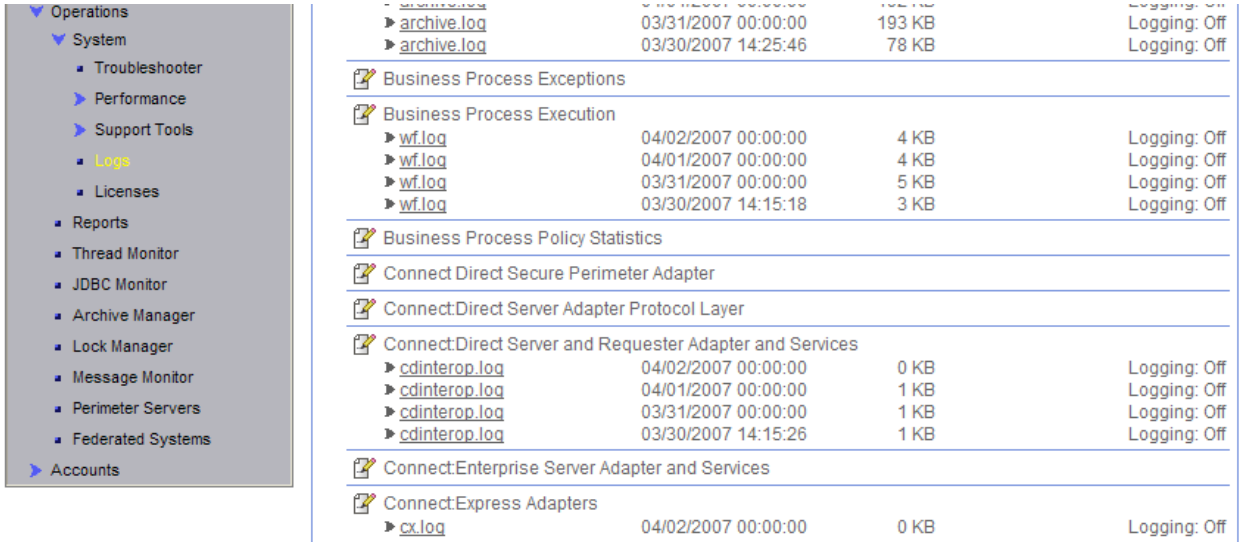
Executing a Connect:Express Service

While executing the business process, a call to a Connect:Express service writes the service request parameters in the process data (<CX*Service*Request> element), then the service response (<CX*Service*Response> element) with the return code, possible error message and any information related to the service (number of elements returned, transfer request number …) .

For GET and LIST modes, the execution of the service writes the information returned in a document CX<service>_<step>. Most are XML documents.

Logging Calls to Services

If you select option Logging=True for a service, the service parameters are recorded in the cx.log file. Use Operations / System / Logs as shown below. Select ConnectExpress Adapters / cxlog to display the trace of call to Connect:Express services.




```
[2007-03-30 17:15:22.222] ALL 000000000000 GLOBAL_SCOPE
Procces:CXPartnerGetAllP.Id:286085.Service:CXPartnerInst.Request:partner
Serverid=localhost:9000:X:ADMIN:****:FALSE:0:30:0,PartnerName=*,Mode=get,

[2007-03-30 17:39:12.647] ALL 000000000000 GLOBAL_SCOPE
Procces:CXTransferSubWaitP.Id:286111.Service:CXTransferSubInst.Request:transfer
Serverid=localhost:9000:X:ADMIN:****:FALSE:0:30:0,FileName=FILE01,PhysicalName=
$TOM_DIR/config/sysin.txt,TransferDirection=T,PartnerName=BOUCLE,TypeOfLink=T,F
ileLabel=lab01,LocalName=BOUCLE,LocalPassword=PSW,TypeOfFile=TV,FileRecordLengt
h=8192,Mode=submit,
```

Tracing Communications with a Monitor

If you need to troubleshoot communication problems between a SI service and a monitor, select option ApiTrace=True to trace TCP/IP session. The data flow of the TCP/IP session is written in a file located in the directory « user.home », and called « cxjai.<Monitor-Address>.<api-port >.trc ».

On Unix, user.home = $HOME.
On Windows, user.home = c:\Document And Settings\username.

You can send this trace file to Sterling Commerce support specialists for analysis.

This chapter provides a detailed description of Connect:Express services.

Reference Guide

The following pages show parameters and examples of process data and documents created for each service.

Tables below show parameters, with data types from the following list:

- S: String, uppercase
- s: String case sensitive
- C: Character
- i: Integer
- L: Long
- B: Boolean (true or false)
- D: Date format AAAA/MM/JJHH:MM:SS or HH:MM

Standard default values for modes CREATE and REPLACE, are *spaces* for types S and s, *spaces* for type c, *zero* for types i and L, *false* for type boolean. When the default value is not the standard one, it is specified.

Unix and Win columns indicate if the parameter is available on this OS, and the maximum length of the parameter if applicable.

The first table shows parameters that are shared by all services, and are required.

Parameters Shared by all Services

Parameters related to connection to monitor and login to monitor as well as trace parameters are listed below. They are required.

Field name	Description	Win	Unix	Data Type
Connect:Express Address	IP address or host name of the monitor	128	128	s
Connect:Express api port	TCP/IP Port of the monitor api server	5	5	i
OSType	Operating system on which the monitor is executing (Unix, Windows)	7	7	S
UserName	User name for loggin to the monitor Use « ADMIN »	8	8	S
UserPassword	User password for loggin to the monitor Use « ADMIN »	8	8	S
Logging	True: service parameters are written in the trace file cx.log of SI	5	5	S
ApiTrace	True: TCP/IP session between SI and the monitor (refer to « Tracing Communications with a monitor »)	5	5	S

CXActivity

CXActivity service enables you to display information about active transfers (GET) or list active transfers (LIST). Information is written in a document.

Active transfers have the following status:

Unix:

- A: Waiting selection
- C: Running
- O: Interrupted
- H: Held
- D: Delayed
- J: Waiting delay
- K: Waiting restart

Windows:

- W: Waiting selection
- T: Selected
- C: Running
- I: Interrupted
- S: Selection error
- R: Automatic restart

Information returned by this service is subject to constant change.

CXActivity Parameters

Field name	Description	Win	Unix	Data Type
Mode	get or list	7	7	S
RequestNumber	Transfer request number(wildcard supported)	12	8	S
TransferDirection	Transfer direction (T:Transmission, R:Reception, *:Both)	1	1	c
PartnerName	Symbolic partner name or '*'	8	8	S
FileName	Symbolic file name or '*'	8	8	S
Requestor	Requestor name (User, process, …) or '*' *	8	8	S
TypeOfUser	Requestor type (E:External, I:Internal, *:Both)	1		c
UserRequestId	Transfer request user identification or '*'	16		s
DocumentFormat	List format: XML, Column, Column with header	-	-	-

Note: setting a parameter to '' is equivalent to omitting the parameter.*

Example

Executing the CXActivity service with (Mode=list,DocumentFormat=ColumnWithHeader) results in the following:

ProcessData

```
Process Data

 Process Name: CXActivityListColP     Instance ID: 289669
 Service Name: CXActivityInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXActivityRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <Mode>list</Mode>
  </CXActivityRequest>
  <CXActivityList_1 SCIObjectID="localhost:9ed927:111b192d865:5bba"/>
  <CXActivityResponse>
    <ActivityCount>4</ActivityCount>
    <ReturnCode>ACTIVITY_LIST_SUCCESSFUL</ReturnCode>
  </CXActivityResponse>
</ProcessData>
```

Document

```
CXActivityList_1

 Process Name: CXActivityListColP     Instance ID: 289669
 Service Name: CXActivityInst
 Document in process data:
  ------------------------------------------------------------------------------
Request       Requestor  FileName  PartnerName  Dir  Status
------------------------------------------------------------------------------
09300006      pga        FILE01    BOUCLE       R    C
09300005      pga        FILE01    BOUCLE       R    C
09300002      pga        FILE01    BOUCLE       T    C
09300001      pga        FILE01    BOUCLE       T    C
```

CXConfiguration

The CXConfiguration service Displays the configuration parameters of a monitor and the content of the asset protection key.

CXConfiguration Parameters

Field name	Description	Win	Unix	Data Type
Mode	Get	7	7	S

Example

Executing the CXConfiguration service results in the following:

ProcessData

```
Process Data

 Process Name: CXConfigurationGetP     Instance ID: 286110
 Service Name: CXConfigurationInst
 <?xml version="1.0" encoding="UTF-8"?>
 <ProcessData>
  <CXConfigurationRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <Mode>get</Mode>
  </CXConfigurationRequest>
  <CXConfigurationGet_1 SCIObjectID="localhost:9ed927:111a2c498f0:-48c2"/>
  <CXConfigurationResponse>
    <ReturnCode>CONFIGURATION_GET_SUCCESSFUL</ReturnCode>
  </CXConfigurationResponse>
</ProcessData>
```

Document

```
CXConfigurationGet_1

 Process Name: CXConfigurationGetP     Instance ID: 286110
 Service Name: CXConfigurationInst
 Document Name: CXConfigurationGet_20070330173728611O_1     Document Store:  Database
 Document in process data:

 <?xml version="1.0" encoding="UTF-8"?>
 <CXConfigurationGet>
     <Configuration>
           <OSType>UNIX</OSType>
           <GMTOffset>7200</GMTOffset>
           <AliasName>UNIX</AliasName>
           <ApiPort>9000</ApiPort>
           <ApiAddress/>
           <BuildDatabaseUsed>true</BuildDatabaseUsed>
           <Dpcsid>FEDORA</Dpcsid>
           <FtpListAllUsed>true</FtpListAllUsed>
           <FtpDefaultFile>FTPFILE</FtpDefaultFile>
           <FtpListenAddress/>
           <FtpListenPort>0</FtpListenPort>
           <StatisticsUsed>false</StatisticsUsed>
           <SyslogUsed>false</SyslogUsed>
           <LogSize>10000</LogSize>
```

```
           <NotificationsUsed>true</NotificationsUsed>
           <RunType>C</RunType>
           <Dpcpsw>PSW</Dpcpsw>
           <SessionTimer>1</SessionTimer>
           <TcpipListenAddress/>
           <TcpipListenPort>6677</TcpipListenPort>
           <TraceUsed>false</TraceUsed>
           <MaxTransfer>6</MaxTransfer>
           <TransferTimer>1</TransferTimer>
           <X25LocalPort/>
           <X25LocalAddress/>
           <ProductInfo>C:E/UNIX 142-1</ProductInfo>
           <SystemInfo>Sysname:Linux.Release:2.6.17-1.2157_FC5.Version:#1 Tue Jul 11 22:55:46
                EDT 2006.Machine:i686.Nodename:localhost.localdomain</SystemInfo>
           <Mode>get</Mode>
           <ExpirationDate>NO-LIMIT</ExpirationDate>
           <ApkeyLine>PRODUCT EXPRESS</ApkeyLine>
           <ApkeyLine>PRODUCT NUMBER 082</ApkeyLine>
           <ApkeyLine>EXPIRATION DATE NO-LIMIT</ApkeyLine>
           <ApkeyLine>OPERATING SYSTEM UNIX</ApkeyLine>
           <ApkeyLine>ALIAS $SSL-LABS LINUX</ApkeyLine>
           <ApkeyLine>PESIT NO-LIMIT</ApkeyLine>
           <ApkeyLine>TCPIP NO-LIMIT</ApkeyLine>
           <ApkeyLine>FTP NO-LIMIT</ApkeyLine>
           <ApkeyLine>ETEBAC3 NO-LIMIT</ApkeyLine>
           <Mode>get</Mode>
     </Configuration>
</CXConfigurationGet>
```

CXEtb3Presentation

The CXEtb3Presentation service enables you to manage the Etebac3 presentation tables of a monitor executing on Windows. An entry is identified by the field PresentationTableId.

Note: For managing Etebac3 presentation tables on Unix use CXPresentation.

CXEtb3Presentation Parameters

Parameters for this service depend on the mode.

Modes create, replace or update

Field name	Description	Win	Unix	Data Type
Mode	create, replace or update	7	7	S
PresentationTableId	Name of the Etebac3 presentation table	50		s
TranslationUsed	Transcoding option (true/false)	5		
TranslationToEbcdic	Name or identification number of the transcoding table used from ASCII to EBCDIC	127		
TranslationToAscii	Name or identification number of the transcoding table used from EBCDIC to ASCII	127		

PresentationTableId parameter is required.

Defaults:

For modes **create** and **replace**, the following non standard defaults apply, if the parameter is omitted: TypeOfCompression=N, TranslationToEbcdic=TOMNT.ASC, TranslationToAscii=TOMNT.ASC.

Example

Executing the CXEtb3Presentation service with (Mode=replace) results in the following:

ProcessData

```
Process Data

 Process Name: WCXEtb3PresentationReplaceP     Instance ID: 289056
 Service Name: CXEtb3Presentation
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXEtb3PresentationRequest>
    <Serverid>10.87.15.92:7000:W:ADMIN:****:FALSE:0:30:0</Serverid>
    <PresentationTableId>Etb3 presentation 1</PresentationTableId>
    <TranslationUsed>TRUE</TranslationUsed>
    <TranslationToEbcdic>XX.ASC</TranslationToEbcdic>
    <TranslationToAscii>YY.EBC</TranslationToAscii>
    <Mode>replace</Mode>
  </CXEtb3PresentationRequest>
  <CXEtb3PresentationResponse>
    <ReturnCode>ETB3PRESENTATION_REPLACE_SUCCESSFUL</ReturnCode>
  </CXEtb3PresentationResponse>
</ProcessData>
```

Mode delete

Field name	Description	Win	Unix	Data Type
Mode	Delete	7	7	S
PresentationTableId	Name of the Etebac3 presentation table	50		s

The PresentationTableId field does not support wildcard.

Modes get or list

Field name	Description	Win	Unix	Data Type
Mode	get ou list	7	7	S
PresentationTableId	Name of the Etebac3 presentation table or wildcard	50		s

If PresentationTableId is omitted, all tables are displayed.

Example
Executing the CXEtb3Presentation service with (Mode=get) results in the following:

ProcessData

```
Process Data

 Process Name: WCXEtb3PresentationGetAllP     Instance ID: 289046
 Service Name: CXEtb3Presentation
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXEtb3PresentationRequest>
    <Serverid>10.87.15.92:7000:W:ADMIN:****:FALSE:0:30:0</Serverid>
    <PresentationTableId/>
    <Mode>get</Mode>
  </CXEtb3PresentationRequest>
  <CXEtb3PresentationGet_1 SCIObjectID="localhost:9ed927:111b192d865:-6d1c"/>
  <CXEtb3PresentationResponse>
    <Etb3PresentationCount>2</Etb3PresentationCount>
    <ReturnCode>ETB3PRESENTATION_GET_SUCCESSFUL</ReturnCode>
  </CXEtb3PresentationResponse>
</ProcessData>
```

Document

```
CXEtb3PresentationGet_1

 Process Name: WCXEtb3PresentationGetAllP     Instance ID: 289046
 Service Name: CXEtb3Presentation
 Document Name: CXEtb3PresentationGet_200704021243289046_1     Document Store:  Database
 Document in process data:
  <CXEtb3PresentationGet>
     <Etb3Presentation>
           <PresentationTableId>ETEBAC3 presentation with translation</PresentationTableId>
           <TranslationUsed>true</TranslationUsed>
           <TranslationToEbcdic>TOMNT.ASC</TranslationToEbcdic>
           <TranslationToAscii>TOMNT.EBC</TranslationToAscii>
     </Etb3Presentation>
     <Etb3Presentation>
           <PresentationTableId>ETEBAC3presentation without translation</PresentationTableId>
           <TranslationUsed>false</TranslationUsed>
           <TranslationToEbcdic/>
           <TranslationToAscii/>
     </Etb3Presentation>
</CXEtb3PresentationGet>
```

CXFile

The CXFile service enables you to manage the symbolic files of a monitor. An entry is identified by the field FileName.

CXFile Parameters

Parameters for this service depend on the mode.

Modes create, replace or update

Field name	Description	Win	Unix	Data Type
Mode	create, replace or update	7	7	S
FileName	Symbolic file name Alphanumeric characters: A-Z,0-9,Space	8	8	S
FileComment	User description of the symbolic file	80		s
FileState	Symbolic file status (E: Enabled, H: Disabled)	1	1	C
TypeOfAllocation	Allocation Type (F: Fixed, D:Dynamic)	1	1	C
FileDirection	Transfer direction (T: Transmission, R: Reception, *: Both	1	1	C
TypeOfFile	Type of file (TF: Text fixed, TV: Text variable, BF: Binary fixed, BU: Binary undefined)	2	2	S
FileOpenOption	File open option (N: New, R: replace, O: Old)	1	1	C
FileSender	Partner or list of partners allowed to transmit the file Characters supported: A-Z,0-9,Space,$,* (Windows) A-Z,0-9, Space,$,* and # (Unix)	8	8	S
FileReceiver	Partner or list of partners allowed to receive the file Characters supported: A-Z,0-9,Space,$,* (Windows) A-Z,0-9, Space,$,* and # (Unix)	8	8	S
PresentationTableId	Name or identification number of the presentation table	50	1	S
LocalPhysicalName	Local physical name	127	44	s
FileRecordLength	Record length	5	5	i
StartTransmitExit	Beginning of transmission user exit	127	12	s(*)
EndTransmitExit	End of transmission user exit	127	12	s(*)
StartReceiveExit	Beginning of reception user exit	127	12	s(*)
EndReceiveExit	End of reception user exit	127	12	s(*)
StartTransmitCommand	Beginning of transmission user command	127	12	s(*)
EndTransmitCommand	End of transmission user command	127	12	s(*)
StartReceiveCommand	Beginning of reception user command	127	12	s(*)
EndReceiveCommand	End of reception user command	127	12	s(*)
ErrorCommand	Transfer error user command	127		s
NotifyUsed	Notification option (true/false)	5		B
ClientToNotify	Name of the client to notify	8		S
Pi99OffsetT	Offset of Pi99ValueT field for pi99 of transmission	3		i
Pi99LengthT	Length of Pi99ValueT field for pi99 of transmission	3		i
Pi99ValueT	Pi99 field for transmission	254		s
Pi99OffsetR	Offset of Pi99ValueR field for pi99 of reception	3		i
Pi99LengthR	Length of Pi99ValueR field for pi99 of reception	3		i
Pi99ValueR	Pi99 field for reception	254		s
FileLabel	User identification of the file	80		s
Priority	Priority (0: High, 1: Normal, 2: Low)		1	C
RemotePhysicalName	Remote physical name		44	S
FtpOptions	FTP Options (Type/Structure/Mode)		3	S

	Type:(E:Ebcdic, A:Ascii, B: Binary, *: unchanged) Structure (F: File, R: Record, *: Unchanged) Mode (B: Block, S: Stream, *: Unchanged)			
ParamFileUsed	Parameter card file option (true/false)		5	B
SpaceAllocationUsed	Space allocation option (true/false)		5	B
FtpStoreUniqueUsed	Ftp « store unique » option (true/false)		5	B
FileAgentUsed	File agent option (true/false)		5	B
TypeOfNotification	Type of Notification: 1 character (‘0’ to ‘7’). ‘0’: No notification. ‘1’: Notification at the beginning of the transfer. ‘2’: Notification at the end of the transfer. ‘4’: Notification if transfer error. Other possibilities are combinations with inclusive « OR » of these values. For example: ‘6’ = ‘2’ OR ‘4’ for a notification at the end of transfer or in case of transfer error. **Windows**: This flag is used for HTTP notification only. **Unix:** This flag is used for HTTP notification or standard notification depending on the values of the keywords HTTPNF and NOTIFY in the sysin configuration file.	1	1	C

With modes **create** and **replace** the following parameters are required:

Unix:
FileName, FileDirection, PresentationTableId, TypeOfFile, FileSender, FileReceiver, LocalPhysicalName.
Windows:
FileName, FileName, FileDirection, PresentationTableId.
if FileDirection=R ou *: FileSender.
if FileDirection=T ou *: FileReceiver and FileRecordLength.
if NotifyUsed=true: ClientToNotify.

With mode **update** the following parameters are required:

FileName.

Defaults:

With modes **create** et **replace,** the following non standard defaults apply, if the parameter is omitted:
Unix,Windows: FileState = E, TypeOfAllocation=D, FileOpenOption=R.
Unix: FtpOptions=”***”.

(*) Supported characters for user exit and user commands fields are: A-Z, 0-9, a-z, Space, $, ., /, & and _ .

Example
Executing the CXFile service with (Mode=replace) results in the following:

ProcessData

```
Process Data

 Process Name: CXFileReplaceP     Instance ID: 286098
 Service Name: CXFileInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXFileRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <FileName>TEST</FileName>
    <FileState>E</FileState>
    <FileComment/>
    <TypeOfAllocation>D</TypeOfAllocation>
    <FileDirection>*</FileDirection>
    <LocalPhysicalName>$TOM_DIR/in/&REQNUMB.tmp</LocalPhysicalName>
```

```
    <Priority>1</Priority>
    <FileOpenOption>R</FileOpenOption>
    <SpaceAllocationUsed>FALSE</SpaceAllocationUsed>
    <FileSender>$$ALL$$</FileSender>
    <FileReceiver>$$ALL$$</FileReceiver>
    <PresentationTableId>1</PresentationTableId>
    <TypeOfFile>TV</TypeOfFile>
    <FileRecordLength>8192</FileRecordLength>
    <StartTransmitCommand/>
    <EndTransmitCommand/>
    <StartReceiveCommand/>
    <EndReceiveCommand>recv.sh</EndReceiveCommand>
    <ErrorCommand/>
    <StartTransmitExit/>
    <EndTransmitExit/>
    <StartReceiveExit/>
    <EndReceiveExit/>
    <FileLabel>lab01</FileLabel>
    <ParamFileUsed>FALSE</ParamFileUsed>
    <Pi99OffsetT/>
    <Pi99LengthT/>
    <Pi99ValueT/>
    <Pi99OffsetR/>
    <Pi99LengthR/>
    <Pi99ValueR/>
    <RemotePhysicalName/>
    <FtpOptions>***</FtpOptions>
    <FtpStoreUniqueUsed>FALSE</FtpStoreUniqueUsed>
    <FileAgentUsed>FALSE</FileAgentUsed>
    <TypeOfNotification>0</TypeOfNotification>
    <NotifyUsed>TRUE</NotifyUsed>
    <ClientToNotify/>
    <Mode>replace</Mode>
  </CXFileRequest>
  <CXFileResponse>
    <ReturnCode>FILE_REPLACE_SUCCESSFUL</ReturnCode>
  </CXFileResponse>
</ProcessData>
```

Mode delete

Field name	Description	Win	Unix	Data Type
Mode	Delete	7	7	S
FileName	Symbolic file name (no wildcard)	8	8	S

FileName field don’t support wildcard.

Modes get or list

Field name	Description	Win	Unix	Data Type
Mode	get ou list	4	4	S
FileName	Symbolic file name or wildcard	8	8	S

If FileName field is omitted, all files are displayed or listed.

Example
Executing the CXFile service with (Mode=get) results in the following:

ProcessData

```
Process Data

 Process Name: CXFileGetAllP     Instance ID: 286097
 Service Name: CXFileInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXFileRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <FileName/>
    <Mode>get</Mode>
  </CXFileRequest>
  <CXFileGet_1 SCIObjectID="localhost:9ed927:111a2c498f0:-4be2"/>
  <CXFileResponse>
    <FileCount>3</FileCount>
    <ReturnCode>FILE_GET_SUCCESSFUL</ReturnCode>
  </CXFileResponse>
</ProcessData>
```

Document

```
CXFileGet_1

 Process Name: CXFileGetAllP     Instance ID: 286097
 Service Name: CXFileInst
 Document Name: CXFileGet_20070330172228609 7_1     Document Store:  Database
 Document in process data:
  <CXFileGet>
     <File>
           <FileName>FILE01</FileName>
           <FileState>E</FileState>
           <TypeOfAllocation>D</TypeOfAllocation>
           <FileDirection>*</FileDirection>
           <TypeOfFile>TV</TypeOfFile>
           <FileOpenOption>R</FileOpenOption>
           <FileSender>$$ALL$$</FileSender>
           <FileReceiver>$$ALL$$</FileReceiver>
           <PresentationTableId>1</PresentationTableId>
           <LocalPhysicalName>$TOM_DIR/in/&REQNUMB.txt</LocalPhysicalName>
           <FileRecordLength>2048</FileRecordLength>
           <StartTransmitExit/>
           <EndTransmitExit/>
           <StartReceiveExit/>
           <EndReceiveExit/>
           <StartTransmitCommand/>
           <EndTransmitCommand/>
           <StartReceiveCommand/>
           <EndReceiveCommand/>
           <Priority>1</Priority>
           <RemotePhysicalName/>
           <FtpOptions>***</FtpOptions>
           <ParamFileUsed>false</ParamFileUsed>
           <SpaceAllocationUsed>false</SpaceAllocationUsed>
           <FtpStoreUniqueUsed>false</FtpStoreUniqueUsed>
           <FileAgentUsed>false</FileAgentUsed>
           <TypeOfNotification>2</TypeOfNotification>
     </File>
     <File>
           <FileName>FILE02</FileName>
           …
     <File>
     …

</CXFileGet>
```

Example
Executing the CXFile service with (Mode=list) results in the following:

ProcessData

```
Process Data

 Process Name: CXFileListP     Instance ID: 286099
 Service Name: CXFileInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXFileRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <FileName/>
    <Mode>list</Mode>
  </CXFileRequest>
  <CXFileList_1 SCIObjectID="localhost:9ed927:111a2c498f0:-4b7b"/>
  <CXFileResponse>
    <FileCount>3</FileCount>
    <ReturnCode>FILE_LIST_SUCCESSFUL</ReturnCode>
  </CXFileResponse>
</ProcessData>
```

Document

```
CXFileList_1

 Process Name: CXFileListP     Instance ID: 286099
 Service Name: CXFileInst
 Document Name: CXFileList_200703301725286099_1     Document Store:  Database
 Document in process data:
  <CXFileList>
     <FileList>
           <FileName>FILE01</FileName>
           <FileName>FILE02</FileName>
           <FileName>TEST</FileName>
     </FileList>
</CXFileList>
```

CXJournal

The CXJournal service enables you to display the Journal records. The journal record identifier is the RequestNumber field.

Note: the amount of data that is received when requesting the list of transfers can be very large; you should consider this problem when setting the selection criteria.

CXJournal Parameters

Modes get or list

Field name	Description	Win	Unix	Data Type
Mode	get ou list	4	4	S
RequestNumber	Transfer request number	12	8	S
Requestor	Requestor name (User, process, …)	8		S
FileName	Symbolic file name	8	8	S
PartnerName	Symbolic partner name	8	8	S
TransferDirection	Transfer direction(T: Transmission, R: Reception, *: Both)	1	1	C
MinimumDate	Selection start date or *	18	18	D
MaximumDate	Selection end date or *	18	18	D
Status	Transfer status or * Unix: A: waiting selection, H: held, K: waiting restart, C:running, E: ended, D: delayed, J: waiting delay, O: interrupted		1	C

Example
Executing the CXJournal service with (Mode=get) results in the following:

ProcessData

```
Process Data

 Process Name: CXJournalGetAllP     Instance ID: 286114
 Service Name: CXJournalInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXJournalRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <Interval>600</Interval>
    <Mode>get</Mode>
  </CXJournalRequest>
  <CXJournalGet_1 SCIObjectID="localhost:9ed927:111a2c498f0:-4746"/>
  <CXJournalResponse>
    <JournalCount>2</JournalCount>
    <ReturnCode>JOURNAL_GET_SUCCESSFUL</ReturnCode>
  </CXJournalResponse>
</ProcessData>
```

Document

```
CXJournalGet_1

 Process Name: CXJournalGetAllP     Instance ID: 286114
 Service Name: CXJournalInst
 Document Name: CXJournalGet_200703301744286114_1     Document Store:  Database
 Document in process data:
  <CXJournalGet>
     <Transfer>
           <RequestNumber>08900007</RequestNumber>
           <TransferIdent>5@0007</TransferIdent>
           <ExternalRequestNumber>08900007</ExternalRequestNumber>
           <Requestor>REMAPI</Requestor>
           <Trc>0000</Trc>
           <TrcMessage/>
           <Prc>0000</Prc>
           <Src>0000</Src>
           <Nrc>0000</Nrc>
           <TcpipRc>0000</TcpipRc>
           <X25Rc/>
           <X25Cause/>
           <X25Diagnostic/>
           <Status>E</Status>
           <StatusMessage>Ended</StatusMessage>
           <TypeOfUser>I</TypeOfUser>
           <TypeOfRequest>N</TypeOfRequest>
           <TransferDirection>T</TransferDirection>
           <Priority>1</Priority>
           <TypeOfLink>T</TypeOfLink>
           <FileName>FILE01</FileName>
           <PhysicalName>$TOM_DIR/config/sysin.txt</PhysicalName>
           <LocalPhysicalName>$TOM_DIR/config/sysin.txt</LocalPhysicalName>
           <RemotePhysicalName/>
           <FileLabel>lab01</FileLabel>
           <TransferOrigin>BOUCLE</TransferOrigin>
           <TransferDestination>BOUCLE</TransferDestination>
           <TransferSender/>
           <TransferReceiver/>
           <FileOrganization>S</FileOrganization>
           <TypeOfFile>T</TypeOfFile>
           <TypeOfCompression>M</TypeOfCompression>
           <TypeOfData>A</TypeOfData>
           <FileRecordLength>2048</FileRecordLength>
           <NetworkMessageSize>4096</NetworkMessageSize>
           <LocalName>BOUCLE</LocalName>
           <PartnerName>BOUCLE</PartnerName>
           <TypeOfPartner>O</TypeOfPartner>
           <Protocol>0</Protocol>
           <CrcUsed>false</CrcUsed>
           <TcpipPort>6677</TcpipPort>
           <TcpipHostName/>
           <TcpipAddress>127.0.0.1</TcpipAddress>
           <X25RemoteAddress/>
           <X25LocalAddress/>
           <X25LocalPort/>
           <X25UserDataField/>
           <X25Facilities/>
           <TransferBeginningDate>2007/03/30 17:39:12</TransferBeginningDate>
           <TransferEndDate>2007/03/30 17:39:13</TransferEndDate>
           <RetryNumber>0</RetryNumber>
           <FileOpenOption>R</FileOpenOption>
           <NetworkBytes>000000000789</NetworkBytes>
           <FileNumberOfRecords>22</FileNumberOfRecords>
           <DateOfExecution>2007/03/30 17:39:12</DateOfExecution>
           <PartnerPassword>PSW</PartnerPassword>
           <UserDataReceived/>
           <UserDataSent/>
           <OriginPhysicalName/>
```

```
           <FileApi/>
           <ProcessId>21258</ProcessId>
           <FileAgentUsed>false</FileAgentUsed>
           <MultiArticleUsed>true</MultiArticleUsed>
           <TranslationTableNumber>0</TranslationTableNumber>
           <TypeOfNotification>2</TypeOfNotification>
           <FtpOptions>***</FtpOptions>
           <FtpStoreUniqueUsed>false</FtpStoreUniqueUsed>
           <SpaceAllocationUsed>false</SpaceAllocationUsed>
           <SessionTimer>0</SessionTimer>
           <TransferTimer>0</TransferTimer>
           <MaxRetries>0</MaxRetries>
           <Mode>get</Mode>
     </Transfer>
     <Transfer>
     …
     </Transfer>
</CXJournalGet>
```

Example
Executing the CXJournal service with (Mode=list,DocumentFormat=ColumnWithHeader) results in the following:

ProcessData

```
Process Data

 Process Name: CXJournalListColP     Instance ID: 289007
 Service Name: CXJournalInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXJournalRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <Mode>list</Mode>
  </CXJournalRequest>
  <CXJournalList_1 SCIObjectID="localhost:9ed927:111b192d865:-797d"/>
  <CXJournalResponse>
    <JournalCount>156</JournalCount>
    <ReturnCode>JOURNAL_LIST_SUCCESSFUL</ReturnCode>
  </CXJournalResponse>
</ProcessData>
```

Document

```
CXJournalList_1

 Process Name: CXJournalListColP     Instance ID: 289007
 Service Name: CXJournalInst
 Document Name: CXJournalList_200704021124289007_1     Document Store:  Database
 Document in process data:
  --------------------------------------------------------------------------------
Request       Requestor  FileName  PartnerName  Dir  Status Date
--------------------------------------------------------------------------------
08700104      pga        FILE01    BOUCLE       T    E      2007/03/28 15:30:35
08700103      pga        FILE01    BOUCLE       T    E      2007/03/28 15:30:31
08700102      pga        FILE01    BOUCLE       R    E      2007/03/28 15:30:44
08700101      pga        FILE01    BOUCLE       T    E      2007/03/28 15:30:44
 …
```

CXPartner

The CXPartner service enables you to manage symbolic partners. An entry is identified by the field PartnerName.

CXPartner Parameters

Parameters for this service depend on the mode.

Modes create, replace or update

Field name	Description	Win	Unix	Data Type
Mode	create, replace or update	7	7	S
PartnerName	Symbolic partner name Supported characters: A-Z,0-9,Space	8	8	S
PartnerPassword	Symbolic partner password Supported characters: A-Z,0-9,Space	8	8	S
LocalNameType	Local identifier type(D:Dynamic, S:Static)	1		C
LocalName	Local alias name of Connect:Express	8	8	s
LocalPassword	Local alias password of Connect:Express	8	8	s
PartnerComment	User description of the symbolic partner	80		S
PartnerState	Status of the partner (E: Enabled, H: Disabled)	1	1	C
TypeOfPartner	Type of the partner (O: Non Connect:Express, T: Connect:Express)	1	1	C
RestartUsed	Automatic restart option (true, false)	5		B
Protocol	File transfer protocol Unix: 1: Etebac3, 2: FTP, 3: PeSIT. Note: On Unix, PeSIT level D ou E is set in the session table Windows: D: PeSIT-D, E: PeSIT-E, 3: Etebac3.	1	1	C
MaxSession	Maximum simultaneous sessions	3	2	i
MaxSessionIn	Maximum simultaneous Incoming sessions	3	2	i
MaxSessionOut	Maximum simultaneous outgoing sessions	3	2	i
SessionTableId	Name or identification number of the session table *Note: Session table number are from « 0 » to « 9 » on Unix*	50	1	S
TypeOfLink	Network link type Unix: T: TCP/IP, X: X25, P: PAD, M: TCP/IP+X25 Windows: T: TCP/IP, S: SNA LU6.2	1	1	C
TcpipAddress	Remote TCP/IP address	15	15	S
TcpipHostName	Remote TCP/IP host name	127	32	s
TcpipPort	Remote TCP/IP port	5	5	i
SnaLuName	Remote SNA LU6.2 address	8		S
AppcModeName	Remote LU6.2 mode name	8		S
AppcTpName	Remote LU6.2 transaction program name	64		S
X25LocalAddress	Local X25 address Numeric characters: 0-9	15	15	S
X25LocalPort	Local X25 port number	2	1	S
X25RemoteAddress	Remote X25 address Numeric characters: 0-9	15	15	S
X25UserDataField	X25 user data field Hexadecimal characters: 0-9,A-F	8	8	S
X25Facilities	X25 facilities Hexadecimal characters: 0-9,A-F	32	16	S
FtpDefaultFile	Default FTP file name Supported characters: A-Z,0-9,Space		8	S

RetryNumber	Maximum number of retries		8	i
SessionTimer	Session retry Timer		2	i
TransferTimer	Transfer retry Timer		2	i
SsslParmId	SSLPARM identifier Supported characters: A-Z,0-9,Space		8	S
RemoteClientSubjectDn	Criteria for remote client subject DN control	256		S
RemoteClientRootDn	Criteria for remote client root DN control	256		S
RemoteServerSubjectDn	Criteria for remote server subject DN control	256		S
RemoteServerRootDn	Criteria for remote server root DN control	256		S

With modes **create** and **replace** the following parameters are required:

PartnerName, TypeOfPartner, Protocol, SessionTableId, TypeOfLink.

Windows:
For TCP/IP, TcpipAddress or TcpipHostName.
For SNA, SnaLuName and AppcModeName.
For X25, X25LocalPort.

With mode **update** the following parameters are required:

PartnerName.

Defaults:
With modes **create** et **replace,** the following non standard defaults apply, if the parameter is omitted:
Unix,Windows: PartnerState = E.
Windows: RestartUsed=true, MaxSession=2 , MaxSessionIn=1, MacSessionOut=1.

(*) Supported characters for user exit and user commands fields are: A-Z, 0-9, a-z, Blank, $, ., /, & and _ .

Mode delete

Field name	Description	Win	Unix	Data Type
Mode	Delete	6	6	S
PartnerName	Symbolic partner name (no wildcard)	8	8	S

PartnerName field don’t support wildcard.

Modes get or list

Field name	Description	Win	Unix	Data Type
Mode	get ou list	4	4	S
PartnerName	Symbolic partner name or wildcard	8	8	S

If the PartnerName field is omitted, all partners are displayed or listed.

Example
Executing the CXPartner service with (Mode=replace) results in the following:

ProcessData

```
Process Data

 Process Name: CXPartnerReplaceP     Instance ID: 286096
 Service Name: CXPartnerInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXPartnerRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <PartnerName>TEST</PartnerName>
    <PartnerPassword>PSW</PartnerPassword>
    <LocalNameType/>
    <LocalName>TEST</LocalName>
    <LocalPassword>PSW</LocalPassword>
    <PartnerComment/>
    <PartnerState>E</PartnerState>
    <TypeOfPartner>O</TypeOfPartner>
    <RestartUsed/>
    <Protocol>E</Protocol>
    <MaxSession>20</MaxSession>
    <MaxSessionIn>10</MaxSessionIn>
    <MaxSessionOut>10</MaxSessionOut>
    <SessionTableId>1</SessionTableId>
    <TypeOfLink>T</TypeOfLink>
    <TcpipAddress>127.0.0.1</TcpipAddress>
    <TcpipPort>6677</TcpipPort>
    <TcpipHostName/>
    <SnaLuName/>
    <AppcModeName/>
    <AppcTpName/>
    <X25LocalAddress/>
    <X25RemoteAddress/>
    <X25LocalPort/>
    <X25UserDataField/>
    <X25Facilities/>
    <FtpDefaultFile/>
    <RetryNumber/>
    <SessionTimer/>
    <TransferTimer/>
    <SslparmId/>
    <Mode>replace</Mode>
  </CXPartnerRequest>
  <CXPartnerResponse>
    <ReturnCode>PARTNER_REPLACE_SUCCESSFUL</ReturnCode>
  </CXPartnerResponse>
</ProcessData>
```

Example
Executing the CXPartner service with (Mode=get) results in the following:

ProcessData

```
Process Data

 Process Name: CXPartnerGetAllP     Instance ID: 286085
 Service Name: CXPartnerInst

 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXPartnerRequest>
```

```
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <PartnerName/>
    <Mode>get</Mode>
  </CXPartnerRequest>
  <CXPartnerGet_1 SCIObjectID="localhost:9ed927:111a2c498f0:-4f88"/>
  <CXPartnerResponse>
    <PartnerCount>3</PartnerCount>
    <ReturnCode>PARTNER_GET_SUCCESSFUL</ReturnCode>
  </CXPartnerResponse>
</ProcessData>
```

Document

```
CXPartnerGet_1

 Process Name: CXPartnerGetAllP     Instance ID: 286085
 Service Name: CXPartnerInst
 Document Name: CXPartnerGet_200703301715286085_1     Document Store:  Database
 Document in process data:
  <CXPartnerGet>
     <Partner>
           <PartnerName>BOUCLE</PartnerName>
           <PartnerPassword>PSW</PartnerPassword>
           <LocalName>BOUCLE</LocalName>
           <LocalPassword>PSW</LocalPassword>
           <PartnerState>E</PartnerState>
           <TypeOfPartner>O</TypeOfPartner>
           <Protocol>3</Protocol>
           <MaxSession>20</MaxSession>
           <MaxSessionIn>10</MaxSessionIn>
           <MaxSessionOut>10</MaxSessionOut>
           <SessionTableId>1</SessionTableId>
           <TypeOfLink>T</TypeOfLink>
           <TcpipAddress>127.0.0.1</TcpipAddress>
           <TcpipPort>6677</TcpipPort>
           <TcpipHostName/>
           <X25LocalAddress/>
           <X25RemoteAddress/>
           <X25LocalPort/>
           <X25UserDataField/>
           <X25Facilities/>
           <FtpDefaultFile/>
           <RetryNumber>0</RetryNumber>
           <SessionTimer>0</SessionTimer>
           <TransferTimer>0</TransferTimer>
           <SslparmId/>
     </Partner>
     <Partner>
     …
     </Partner>
</CXPartnerGet>
```

Example
Executing the CXPartner service with (Mode=list) results in the following:

ProcessData

```
Process Data

 Process Name: CXPartnerListP     Instance ID: 286095
 Service Name: CXPartnerInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXPartnerRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
```

```
    <PartnerName/>
    <Mode>list</Mode>
  </CXPartnerRequest>
  <CXPartnerList_1 SCIObjectID="localhost:9ed927:111a2c498f0:-4c4b"/>
  <CXPartnerResponse>
    <PartnerCount>3</PartnerCount>
    <ReturnCode>PARTNER_LIST_SUCCESSFUL</ReturnCode>
  </CXPartnerResponse>
</ProcessData>
```

Document

```
CXPartnerList_1

 Process Name: CXPartnerListP     Instance ID: 286095
 Service Name: CXPartnerInst
 Document Name: CXPartnerList_200703301719286095_1     Document Store:  Database
 Document in process data:
  <CXPartnerList>
     <PartnerList>
           <PartnerName>BOUCLE</PartnerName>
           <PartnerName>BOUCLE1</PartnerName>
           <PartnerName>BOUCLE2</PartnerName>
     </PartnerList>
</CXPartnerList>
```

CXPresentation

The CXPresentation service enables you to manage the PeSIT and Etebac3 presentation tables of the Connect:Express Unix monitor and PeSIT presentation tables of the Connect:Express Windows monitor. An entry is identified by the parameter PresentationTableId.
On Unix the presentation tables are identified by numbers « 0 » to « 9 ». You can neither create nor delete a presentation table. On Windows presentation tables are identifed by a string. You can create and delete them

Note: To manage Etebac3 presentation tables of the Connect:Express Windows monitors, use « CXEtb3Presentation » service.

CXPresentation Parameters

Parameters for this service depend on the mode.

Modes create (Windows), replace or update

Field name	Description	Win	Unix	Data Type
Mode	create (Windows only), replace, update	7	7	S
PresentationTableId	Name or identification number of the presentation table	50	1	s
TypeOfCompression	Type of compression (N: none, H: horizontal, V: vertical, M: horizontal and vertical)	C	C	C
ConcatenationUsed	PeSIT FPDU Concatenation option (true/false)	5		B
MultiArticleUsed	PeSIT fpdu Multi-article option (true/false)	5	5	B
SegmentationUsed	PeSIT fpdu Segmentation option (true/false)	5		B
TranslationUsed	Transcoding option (true/false)	5		B
TranslationToEbcdic	Name or identification number of the transcoding table from ASCII to EBCDIC	127	1	S
TranslationToAscii	Name or identification number of the transcoding table from EBCDIC to ASCII	127		S

With modes **create** and **replace** the following parameters are required:

Unix replace:
PresentationTableId, TypeOfCompression, TranslationToEbcdic.
Windows:
PresentationTableId.

With mode **update** the following parameters are required:

PresentationTableId.

Defaults:
With modes **create** et **replace,** the following non standard defaults apply, if the parameter is omitted:
TypeOfCompression=N,TranslationToEbcdic=TOMNT.ASC,TranslationToAscii=TOMNT.ASC (Windows).

Mode delete (Windows)

Field name	Description	Win	Unix	Data Type
Mode	delete (Windows uniquement)	6		S
PresentationTableId	Name or identification number of the presentation table (no wildcard)	50		s

PresentationTableId field does not support wildcard.

Modes get or list

Field name	Description	Win	Unix	Data Type
Mode	get, ou list	4	4	S
PresentationTableId	Name or identification number of the presentation table	50		s

If the PresentationTableId field is omitted, all tables are displayed or listed.

Example
Executing the CXPresentation service with (Mode=replace) results in the following:

ProcessData

```
Process Data

 Process Name: CXPresentationReplaceP     Instance ID: 286109
 Service Name: CXPresentationInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXPresentationRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <PresentationTableId>9</PresentationTableId>
    <TypeOfCompression>M</TypeOfCompression>
    <ConcatenationUsed/>
    <MultiArticleUsed>FALSE</MultiArticleUsed>
    <SegmentationUsed/>
    <TranslationUsed/>
    <TranslationToEbcdic>4</TranslationToEbcdic>
    <TranslationToAscii/>
    <Mode>replace</Mode>
  </CXPresentationRequest>
  <CXPresentationResponse>
    <ReturnCode>PRESENTATION_REPLACE_SUCCESSFUL</ReturnCode>
  </CXPresentationResponse>
</ProcessData>
```

Example
Executing the CXPresentation service with (Mode=get) results in the following:

ProcessData

```
Process Data

 Process Name: CXPresentationGetAllP     Instance ID: 286106
 Service Name: CXPresentationInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXPresentationRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <PresentationTableId/>
    <Mode>get</Mode>
  </CXPresentationRequest>
  <CXPresentationGet_1 SCIObjectID="localhost:9ed927:111a2c498f0:-49df"/>
  <CXPresentationResponse>
    <PresentationCount>9</PresentationCount>
    <ReturnCode>PRESENTATION_GET_SUCCESSFUL</ReturnCode>
  </CXPresentationResponse>
</ProcessData>
```

Document

```
CXPresentationGet_1

 Process Name: CXPresentationGetAllP     Instance ID: 286106
 Service Name: CXPresentationInst
 Document Name: CXPresentationGet_200703301734286106_1     Document Store:  Database
 Document in process data:
  <CXPresentationGet>
     <Presentation>
           <PresentationTableId>1</PresentationTableId>
           <TypeOfCompression>M</TypeOfCompression>
           <MultiArticleUsed>true</MultiArticleUsed>
           <TranslationToEbcdic>0</TranslationToEbcdic>
     </Presentation>
     <Presentation>
     …
     </Presentation>
</CXPresentationGet>
```

CXSession

The CXSession service enables you to manage the session tables of the Connect:Express monitors. An entry is identified by the parameter SessionTableId.
On Unix the session tables are identified by numbers « 0 » to « 9 ». You can neither create nor delete a session table. On Windows session tables are identified by a string. You can create and delete them

CXSession Parameters

Parameters for this service depend on the mode.

Modes create, replace or update

Field name	Description	Win	Unix	Data Type
Mode	create, replace or update	7	7	S
SessionTableId	Name or identification number of the session table	50	1	S
BaseMessageSize	Network message size (negotiated) Windows: 256-4096, Unix: 0-6535	4	5	i
BaseSynchronizationSize	Synchronization buffer size in Kbytes (negotiated) Windows: 1-32, Unix: 0-99	2	2	i
SessionDirection	Direction (T: Transmission only, R: Reception only, *: Both)	1		C
CrcUsed	CRC option (true/false)	5	5	B
ResynchronizationNumber	Number of online retries	2		i
BaseWindowSize	Synchronization window size (negotiated) 0-16		2	i
ProtocolVersion	Protocole version (1: PeSIT-D,2: PeSIT-E)		1	C
RetryNumber	Number of retries		2	i

With modes **create** and **replace** the following parameters are required:

Unix replace: SessionTableId, TypeOfCompression, TranslationToEbcdic.
Windows: SessionTableId.

With mode **update** the following parameters are required:

SessionTableId.

Defauts:

With mode **update,** the following non standard defaults apply, if the parameter is omitted:
ProtocolVersion=2 (Unix).

Modes delete (Windows)

Field name	Description	Win	Unix	Data Type
Mode	Delete	6		S
SessionTableId	Name or identification number of the session table (no wildcard)	50		S

SessionTableId field does not support wildcard.

Modes get or list

Field name	Description	Win	Unix	Data Type
Mode	get ou list	4	4	S
SessionTableId	Name or identification number of the session table	50	1	S

If the SessionTableId field is omitted, all tables are displayed or listed.

Example
Executing the CXSession service with (Mode=replace) results in the following:

ProcessData

```
Process Data

 Process Name: CXSessionReplaceP     Instance ID: 286105
 Service Name: CXSessionInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXSessionRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <SessionTableId>9</SessionTableId>
    <BaseMessageSize>24000</BaseMessageSize>
    <BaseSynchronizationSize>80</BaseSynchronizationSize>
    <SessionDirection/>
    <CrcUsed>TRUE</CrcUsed>
    <ResynchronizationNumber/>
    <BaseWindowSize>8</BaseWindowSize>
    <ProtocolVersion>1</ProtocolVersion>
    <RetryNumber>30</RetryNumber>
    <Mode>replace</Mode>
  </CXSessionRequest>
  <CXSessionResponse>
    <ReturnCode>SESSION_REPLACE_SUCCESSFUL</ReturnCode>
  </CXSessionResponse>
</ProcessData>
```

Example
Executing the CXSession service with (Mode=get) results in the following:

ProcessData

```

```

Document

```
CXSessionGet_1

 Process Name: CXSessionGetAllP     Instance ID: 286104
 Service Name: CXSessionInst
 Document Name: CXSessionGet_200703301732286104_1     Document Store:  Database
 Document in process data:
  <CXSessionGet>
     <Session>
           <SessionTableId>1</SessionTableId>
           <BaseMessageSize>4096</BaseMessageSize>
           <BaseSynchronizationSize>32</BaseSynchronizationSize>
           <CrcUsed>false</CrcUsed>
```

```
            <BaseWindowSize>2</BaseWindowSize>
            <ProtocolVersion>2</ProtocolVersion>
            <RetryNumber>0</RetryNumber>
     </Session>
     <Session>
     …
     </Session>
</CXSessionGet>
```

CXStatistics

If the Statistics function is enabled in the monitor configuration, the CXStatistics service enables you to display selected statistics.

Note: the amount of data that is received when requesting the statistics can be very large; you should consider this problem when setting the selection criteria.

CXStatistics Parameters

Field name	Description	Win	Unix	Data Type
Mode	Get	3	3	S
MinimumDate	Selection start date or *	18	18	D
MaximumDate	Selection end date or *	18	18	D
RequestNumber	Request number or *	12	8	S
Interval	Last events interval in number of seconds	12	12	i

Example
Executing the CXStatistics service with (Mode=get) results in the following:

ProcessData

```
Process Data

 Process Name: CXStatisticsGetP     Instance ID: 286125
 Service Name: CXStatisticsInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXStatisticsRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <Interval>3600</Interval>
    <Limit>100</Limit>
    <Mode>get</Mode>
  </CXStatisticsRequest>
  <CXStatisticsGet_1 SCIObjectID="localhost:9ed927:111a2c498f0:-448b"/>
  <CXStatisticsResponse>
    <StatisticsCount>17</StatisticsCount>
    <ReturnCode>STATISTICS_GET_SUCCESSFUL</ReturnCode>
  </CXStatisticsResponse>
</ProcessData>
```

Document

```
CXStatisticsGet_1

 Process Name: CXStatisticsGetP     Instance ID: 286125
 Service Name: CXStatisticsInst
 Document Name: CXStatisticsGet_200703301753286125_1     Document Store:  Database
 Document in process data:
  <CXStatisticsGet>
     <Statistics>
           <DateOfStatisticsRecord>2007/03/3017:39:12</DateOfStatisticsRecord>
           <RecordId>SB</RecordId>
           <MessageId>UXOL134I</MessageId>
           <MonitorId>FEDORA</MonitorId>
           <OperatingSystem>Unix</OperatingSystem>
           <RequestNumber>08900008</RequestNumber>
           <Status/>
           <Event>ConnectionStarted</Event>
           <MessageText>Communication started - COMMUNICATION OPENED (I) WITH: BOUCLE
               REQ: 08900008 PESIT   TCPIP</MessageText>
           <PartnerName>BOUCLE</PartnerName>
           <LocalName>BOUCLE</LocalName>
           <TypeOfPartner>O</TypeOfPartner>
           <Protocol>E</Protocol>
           <TypeOfLink>T</TypeOfLink>
           <TcpipAddress>127.0.0.1</TcpipAddress>
           <TcpipPort>06677</TcpipPort>
           <RetryNumber>0</RetryNumber>
     </Statistics>
     <Statistics>
     …
     </Statistics>
</CXStatisticsGet>
```

CXTransferSubmit

The CXTransferSubmit service enables you to request a file transfer to a monitor.

You can check transfer completion in several ways:

- Wait for the end of transfer in the CXTransferSubmit service.
- Check the end of transfer after completion of CXTransferSubmit, using either CXTransferStatus service or CXTransferJournal service.
- Set up a http server in SI to receive end of transfer notifications sent by the monitor.

Use the following parameters to indicate to the service how to return result:

End Transfer Condition Parameters

Field name	Description	Win	Unix	Data Type
WaitForTermination	True: The service waits for transfer completion by inquiring the monitor MaxPollCount time, evry PollInterval seconds. False: The service returns immediately after submitting the request.	5	5	S
MaxPollCount	Maximum number of status inquiry before returning (if WaitForTermination=True)	3	3	i
PollInterval	Interval in second between to status inquiries (If WaitForTermination=True)	4	4	i

Use the following parameters to request the transfer to the monitor:

CXTransferSubmit Parameters

Field name	Description	Win	Unix	Data Type
FileName	Symbolic file name	8	8	S
PhysicalName	Physical file name	127	44	S
TransferDirection	Transfer direction (T: transmission, R: reception)	1	1	C
PartnerName	Symbolic partner name	8	8	S
Priority	Transfer priority (0: high, 1: normal, 2: low)	1	1	C
TypeOfRequest	Type of request (N: normal, I: inquiry, H: hold) If TransferDirection=T the type cannot be « I »	1	1	C
TypeOfLink	Network link type Unix: T: TCP/IP, X: X25, P: PAD, M: TCP/IP+X25 Windows: T: TCP/IP, S: SNA LU6.2	1	1	C
Requestor	Name of the requestor (user, process, …)	8		S
UserRequestId	User identification of the request	16		S
LocalName	Local alias name of the monitor	8	8	S
LocalPassword	Local alias password of the monitor	8	8	S
TypeOfFile	Type of file (TF: text fixed, TV: text variable, BF: binary fixed, BU: binary undefined)		2	S
FileRecordLength	Record length If TransferDirection=T, cannot be NULL		5	I
FileApi	User description of the transfer (Etebac3 Card)	80	88	S
FileLabel	User Identification of the file	80	80	S
DateOfExecution	Scheduling date of the request	18	18	D

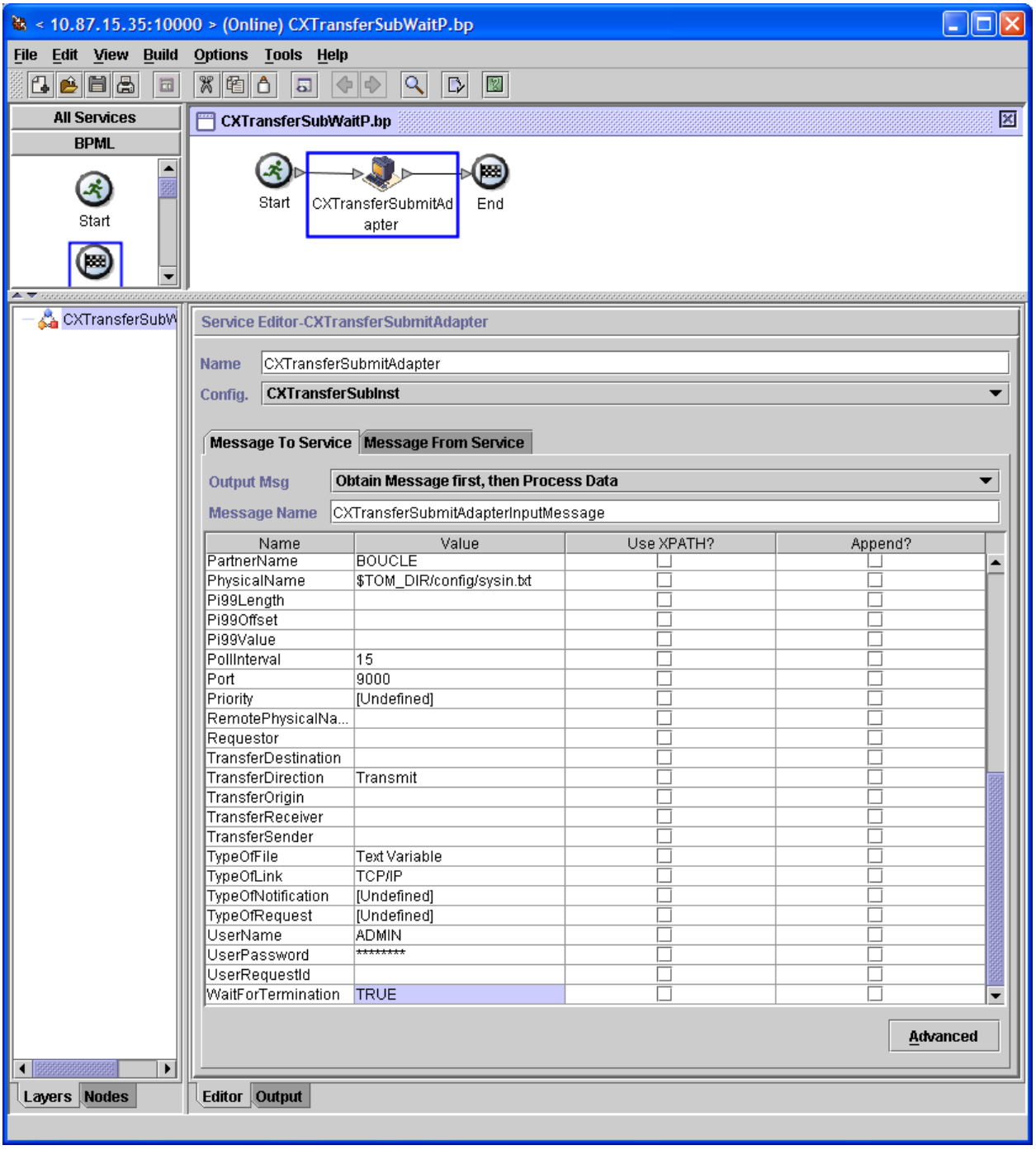
TypeOfNotification	Type of Notification: 1 character (‘0’ to ‘7’). ‘0’: No notification. ‘1’: Notification at the beginning of the transfer. ‘2’: Notification at the end of the transfer. ‘4’: Notification if transfer error. Other possibilities are combinations with inclusive « OR » of these values. For example: ‘6’ = ‘2’ OR ‘4’ for a notification at the end of transfer or in case of transfer error. **Windows**: This flag is used for HTTP notification only. **Unix:** This flag is used for HTTP notification or standard notification depending on the values of the keywords HTTPNF and NOTIFY in the sysin configuration file.	1	1	C
NotifyUsed	Notification option (true/false)	5		B
ClientToNotify	Name of the client to notify Required if NotifyUsed=true	255		S
FtpStoreUniqueUsed	FTP « store unique » option (true/false)		5	B
FtpOptions	FTP Options (Type/Structure/Mode) Type:(E:Ebcdic, A:Ascii, B: Binary, *: unchanged) Structure (F: file, R: record, *: unchanged) Mode (B: bock, S: stream, *: unchanged)		3	S
Pi99Offset	Offset in Pi99Value for pi99	3		I
Pi99Length	Length in Pi99Value forPi99	3		I
Pi99Value	pi99 information	254	254	S
AdHocUser	Remote AdHoc user identification	8		S
AdHocPassword	Remote AdHoc user password	8		S
OriginPhysicalName	AdHoc Local physical name sent	44	44	S
RemotePhysicalName	AdHoc remote physical name sent	44	44	S
TransferOrigin	Transfer requestor	8	8	S
TransferDestination	Transfer addressee	8	8	S
TransferSender	File sender (Pi61)	24	24	S
TransferReceiver	File receiver (Pi62)	24	24	S
FileAgentUsed	File agent option (true/false)		5	B

The following parameters are required:

FileName, TransferDirection, TypeOfRequest.

Example 1

The business process shown below requests a file transfer. The CXTransferSubmit service waits for the transfer completion; it returns successfully if the transfer is ended or with error code after having checked for completion three times, every 15 seconds.



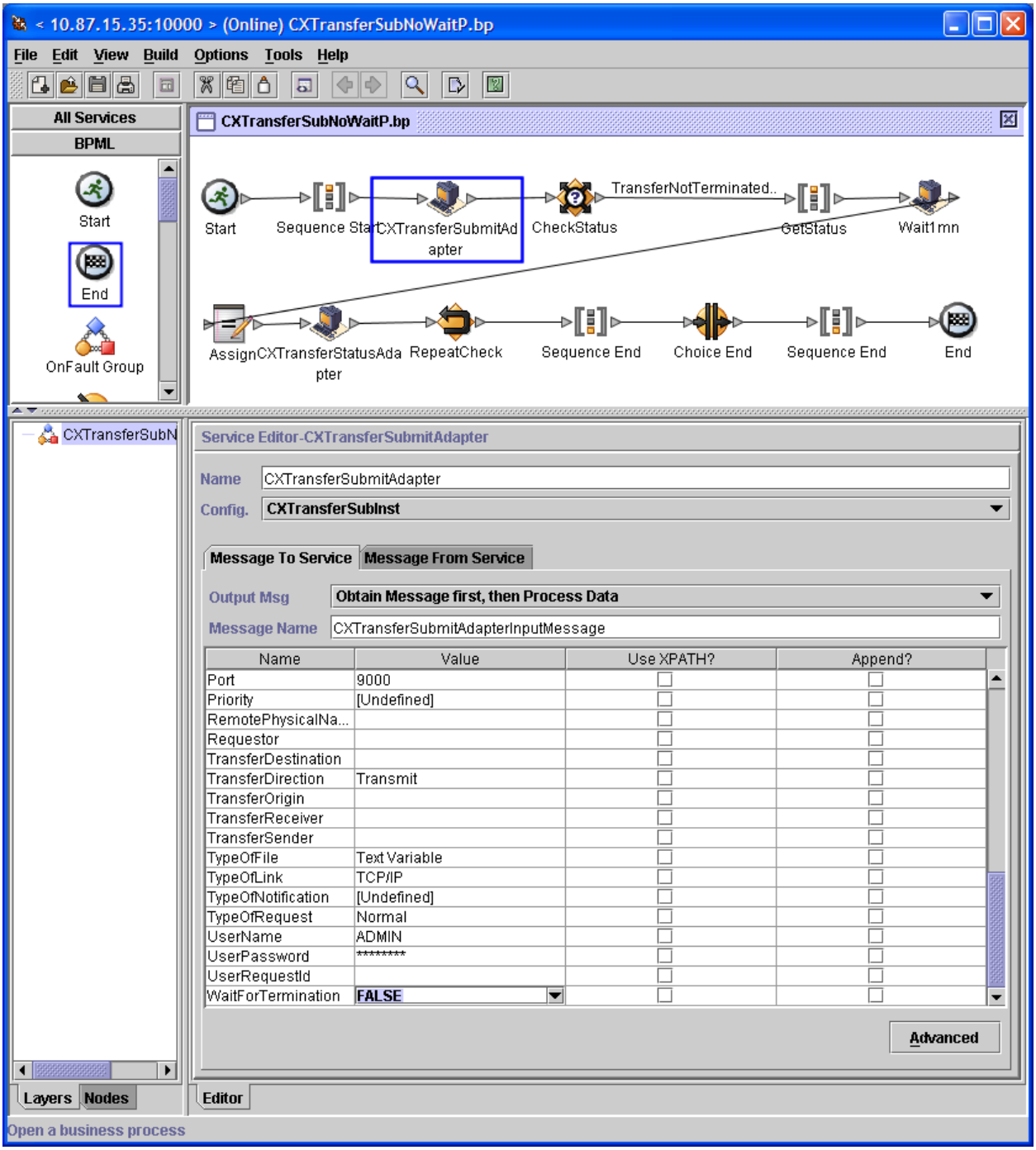
Executing the CXTransferSubmit service results in the following:

ProcessData

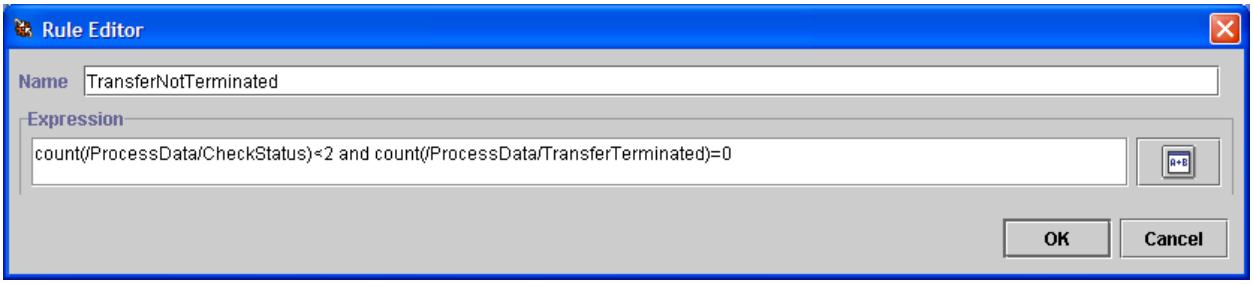
```
Process Data

 Process Name: CXTransferSubWaitP     Instance ID: 286111
 Service Name: CXTransferSubInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXTransferSubmitRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <FileName>FILE01</FileName>
    <PhysicalName>$TOM_DIR/config/sysin.txt</PhysicalName>
    <TransferDirection>T</TransferDirection>
    <PartnerName>BOUCLE</PartnerName>
    <TypeOfLink>T</TypeOfLink>
    <FileLabel>lab01</FileLabel>
    <LocalName>BOUCLE</LocalName>
    <LocalPassword>PSW</LocalPassword>
    <TypeOfFile>TV</TypeOfFile>
    <FileRecordLength>8192</FileRecordLength>
    <Mode>submit</Mode>
    <WaitForTermination>TRUE</WaitForTermination>
    <MaxPollCount>3</MaxPollCount>
    <PollInterval>15</PollInterval>
  </CXTransferSubmitRequest>
  <CXTransferSubmitResponse>
    <ReturnCode>TRANSFER_SUBMIT_SUCCESSFUL</ReturnCode>
    <RequestNumber>08900007</RequestNumber>
  </CXTransferSubmitResponse>
  <CXTransferStatus>
    <ReturnCode>TRANSFER_SUCCESSFUL.Request number 08900007</ReturnCode>
  </CXTransferStatus>
</ProcessData>
```

Example 2

The business process shown below requests a file transfer. The CXTransferSubmit service returns successfully with the request number, without checking for transfer completion. The end of transfer must be checked by mean of CXTransferStatus service.



The TransferNotTerminated condition is defined as shown below:



Executing the CXTransferSubmit service results in the following:

ProcessData

```
Process Data

 Process Name: CXTransferSubNoWaitP     Instance ID: 289079
 Service Name: DecisionEngineService
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXTransferSubmitRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <FileName>FILE01</FileName>
    <PhysicalName>$TOM_DIR/config/sysin.txt</PhysicalName>
    <TransferDirection>T</TransferDirection>
    <PartnerName>BOUCLE</PartnerName>
    <TypeOfLink>T</TypeOfLink>
    <TypeOfRequest>N</TypeOfRequest>
    <FileLabel>lab01</FileLabel>
    <LocalName>BOUCLE</LocalName>
    <LocalPassword>PSW</LocalPassword>
    <TypeOfFile>TV</TypeOfFile>
    <FileRecordLength>8192</FileRecordLength>
    <Mode>submit</Mode>
    <WaitForTermination/>
    <MaxPollCount/>
    <PollInterval/>
  </CXTransferSubmitRequest>
  <CXTransferSubmitResponse>
    <ReturnCode>TRANSFER_SUBMIT_SUCCESSFUL</ReturnCode>
    <RequestNumber>09200003</RequestNumber>
  </CXTransferSubmitResponse>
  <RequestNumber>09200003</RequestNumber>
  <CXTransferStatusRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <RequestNumber>09200003</RequestNumber>
    <Mode>list</Mode>
  </CXTransferStatusRequest>
  <TransferTerminated>TRANSFER_SUCCESSFUL</TransferTerminated>
</ProcessData>
```

CXTransferControl

The CXTransferControl service enables you to interrupt, restart or purge file transfers. For Connect:Express Windows you can only purge active transfers (CHK file). For Connect:Express unix you can purge any transfers because they are all recorded in the RENC file.

CXTransferControl Parameters

Parameters for this service depend on the mode.

Modes interrupt or restart

Field name	Description	Win	Unix	Data Type
Mode	Interrupt ou restart	9	9	S
RequestNumber	Request number (no wildcard)	12	8	S

Mode purge (Windows)

Field name	Description	Win	Unix	Data Type
Mode	Purge	9	9	S
RequestNumber	Request number (no wildcard)	12	8	S

Mode purge (Unix)

Field name	Description	Win	Unix	Data Type
Mode	Purge	6	6	S
FileName	Symbolic file name or *	8	8	S
PartnerName	Symbolic partner name or *	8	8	S
TransferDirection	Transfer direction (T: transmission, R: reception, *: Both)	1	1	C
RequestNumber	Request number (only * is supported)	12	8	S
Status	Transfer status or *		1	S
NotAfter	End purge date or *		18	D
DaysToKeep	Number of days to keep *Note: Today's date is included*		3	S

Example
Executing the CXTransferControl service with (Mode=purge) results in the following:

ProcessData

```
Process Data

 Process Name: CXTransferPurgeP     Instance ID: 289036
 Service Name: CXTransferControlInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXTransferControlRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <DaysToKeep>1</DaysToKeep>
```

```
    <Mode>purge</Mode>
  </CXTransferControlRequest>
  <CXTransferControlResponse>
    <ReturnCode>TRANSFERCONTROL_PURGE_SUCCESSFUL</ReturnCode>
  </CXTransferControlResponse>
</ProcessData>
```

CXTransferStatus

The CXTransferStatus service enables you to show in the « process data », the status of a transfer (ended, running). You can use it in a business process as a condition.

Depending on the status of the transfer, the following information is shown in the ProcessData:

- <CheckStatus>TRANSFER_IN_PROGESS</CheckStatus>
- <TransferTerminated>TRANSFER_SUCCESSFULL</TransferTerminated>
- <TransferTerminated>TRANSFER_FAILED</TransferTerminated>

CXTransferStatus Parameters

Field name	Description	Win	Unix	Data Type
RequestNumber	Transfer request number	12	8	S

Example
Executing CXTransferStatus service after CXTransferSubmit service results in the following, in case of transfer successful completion:

ProcessData

```
Process Data

 Process Name: CXTransferSubNoWaitP     Instance ID: 289037
 Service Name: DecisionEngineService
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CXTransferSubmitRequest>
     ...
    <WaitForTermination/>
    <MaxPollCount/>
    <PollInterval/>
  </CXTransferSubmitRequest>
  <CXTransferSubmitResponse>
    <ReturnCode>TRANSFER_SUBMIT_SUCCESSFUL</ReturnCode>
    <RequestNumber>09200001</RequestNumber>
  </CXTransferSubmitResponse>
  <RequestNumber>09200001</RequestNumber>
  <CXTransferStatusRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <RequestNumber>09200001</RequestNumber>
    <Mode>list</Mode>
  </CXTransferStatusRequest>
  <TransferTerminated>TRANSFER_SUCCESSFUL</TransferTerminated>
</ProcessData>
```

This chapter describes how you can receive Connect:Express notifications in SI.

Notifications

You can configure a Connect:Express monitor to send file transfer notifications, at the beginning or at end of transfer. The monitor sends the notification via HTTP, to a SI server that can be configured to launch a business process.

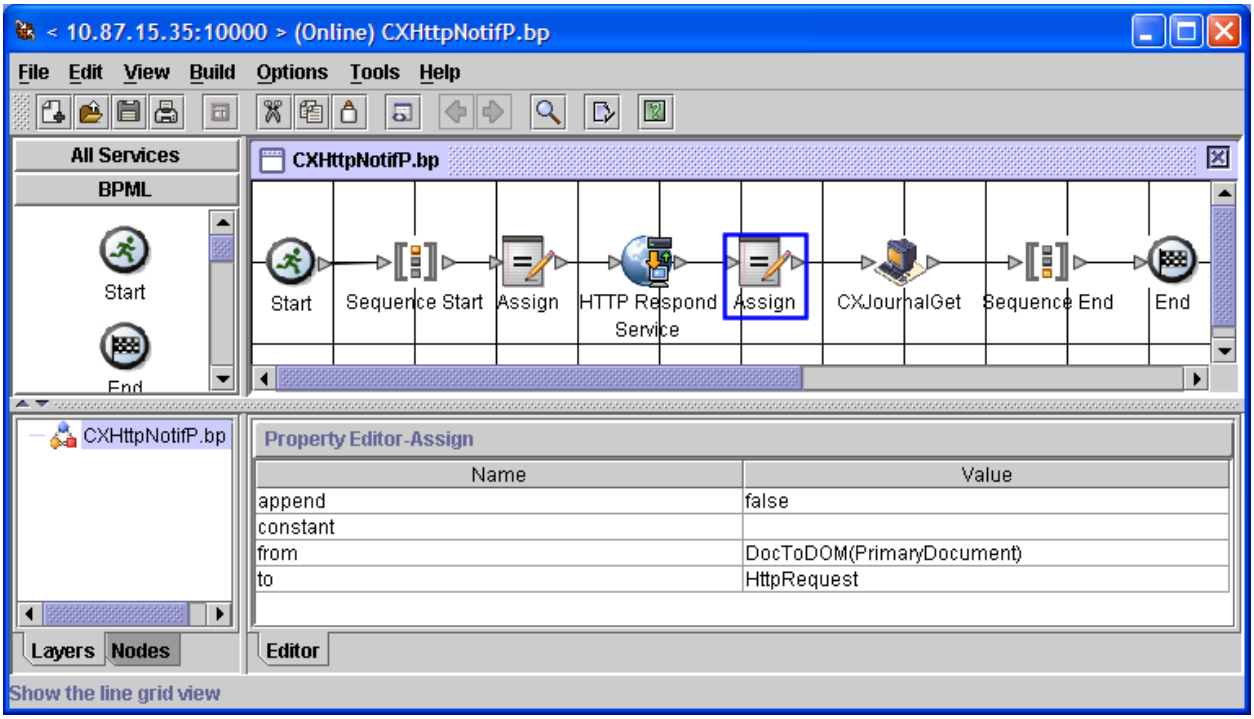
Configuring a SI http Server to Receive Notifications

You need to create a business process that will process the notification and to configure the http server to activate the business process.

Creating the Business Process to Receive the Notification

The business process shown bellow executes like this:

- The « HTTP Respond Service » service writes the Notification (http request body) in the « primary document ». The notification contains the transfer request number.
- The « Assign » service copies the notification in <HttpRequest> element of the process data.
- Le « CXJournalGet » service uses the request number to inquire the corresponding Journal record to the monitor.



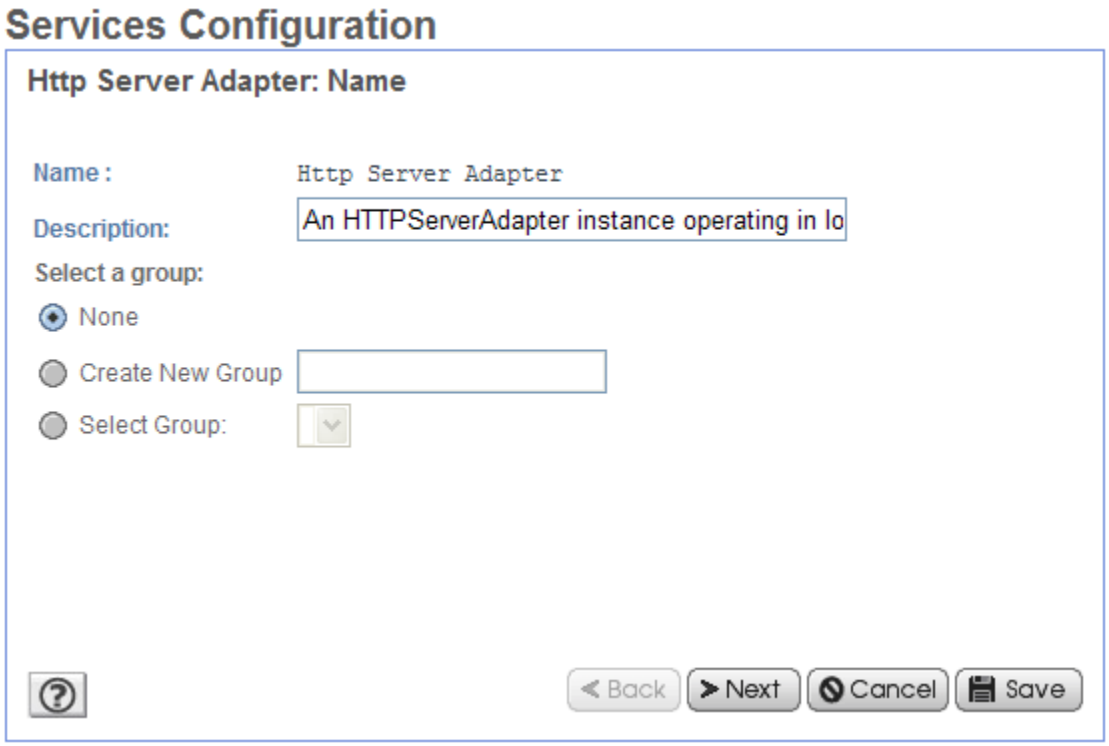
Associating the Business Process to an URI of the http Server

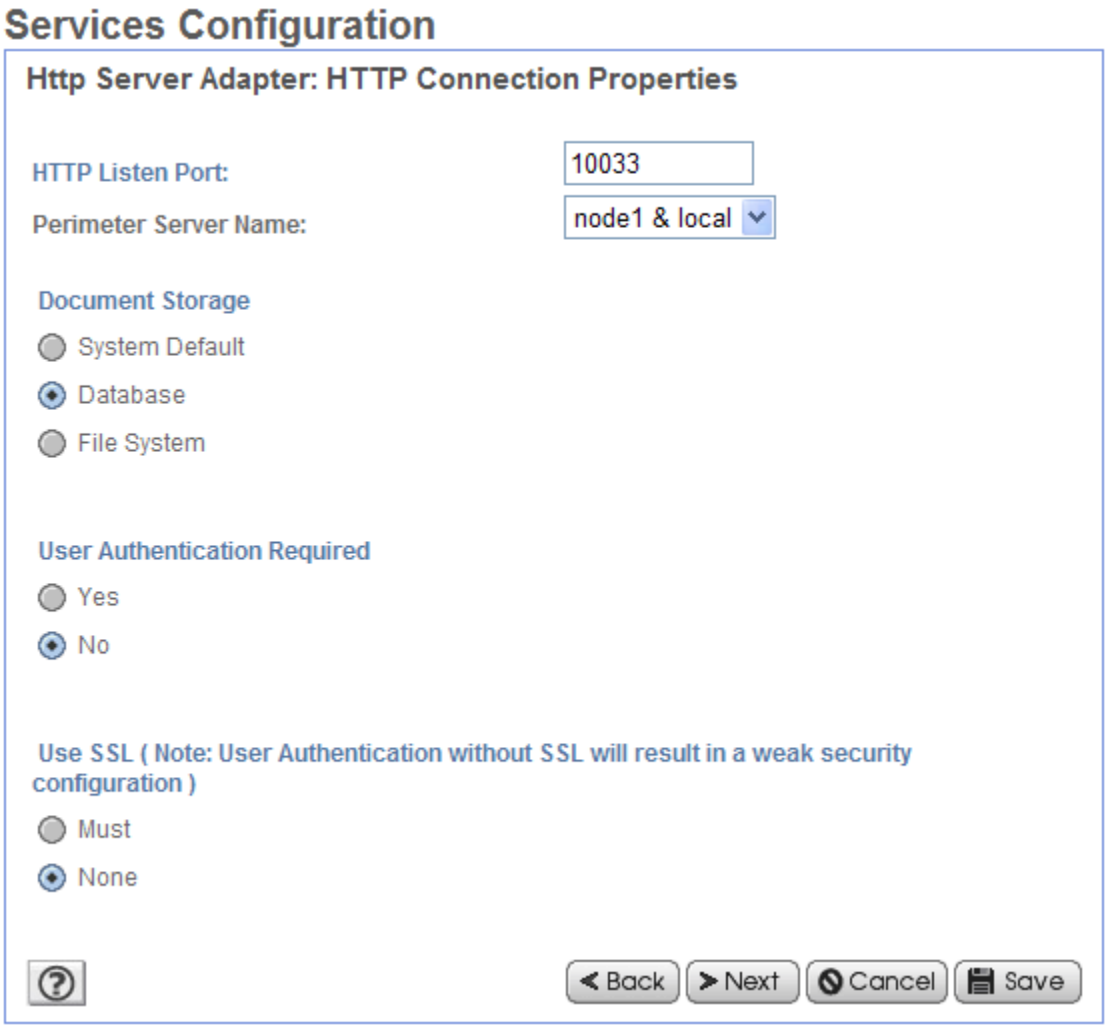
Connect:Express sends the notifications to port **10033** of the SI HTTP server, to URI « **/cxnotif** », as shown before. You can however change these parameters (create a new HTTP server service Instance on a different port, a different URI).

Use SI interface (Deployment/Services/Configuration) to add URI « /cxnotif » to the http server instance that is already defined, and associate to this URI the business process « CXHttpNotif » shown in the previous paragraph.

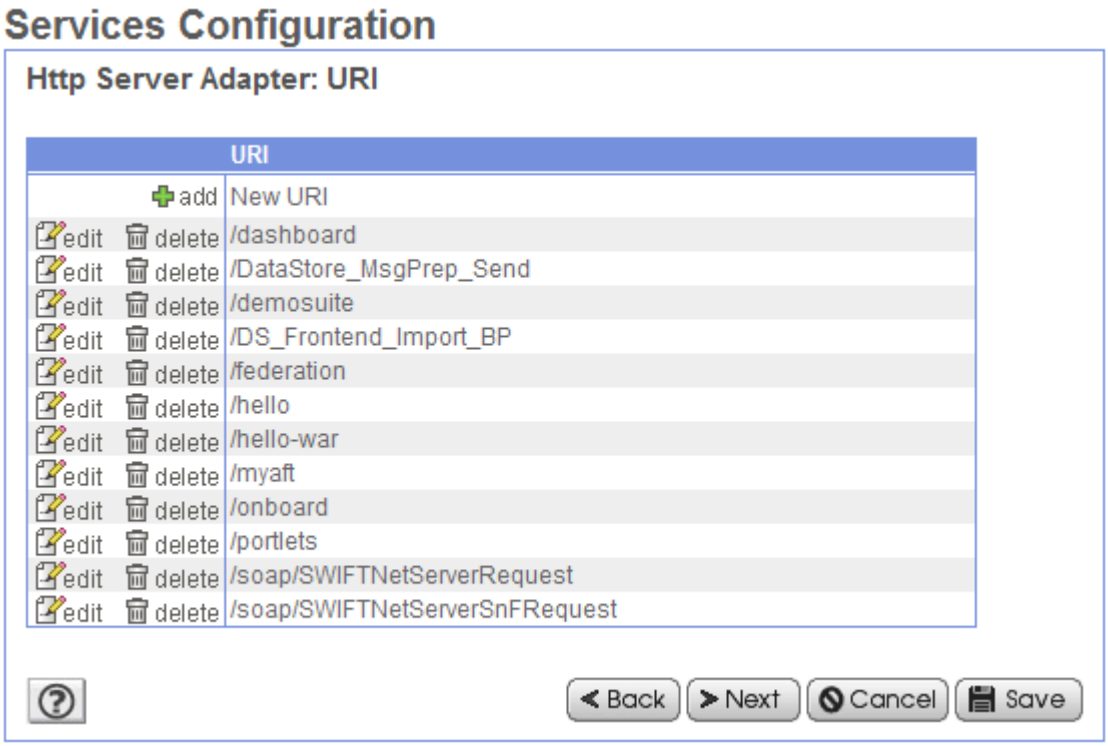
*Note: Connect:Express monitors don't implement **SSL, and user authentication** in the http request.*

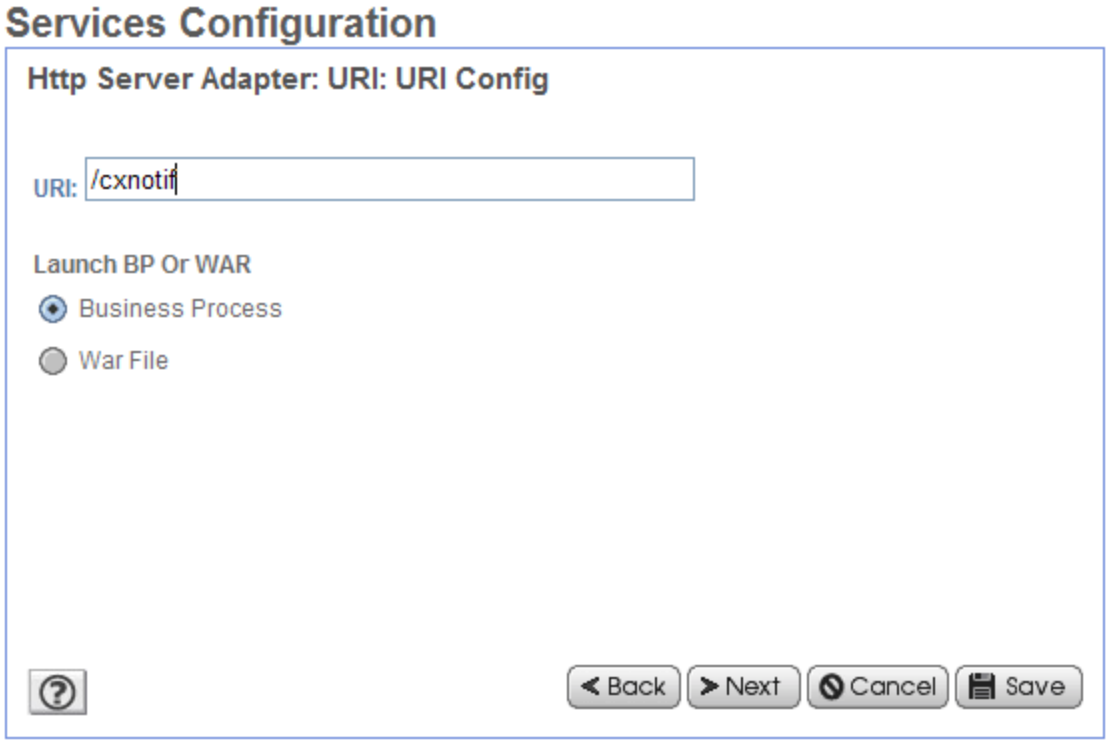
Configuration is done as shown in the following:


Services Configuration
Http Server Adapter: Name
Name :
Http Server Adapter
Description:
An HTTPServerAdapter instance operating in lo
Select a group:
None
Create New Group
Select Group:
Back
Next
Cancel
Save

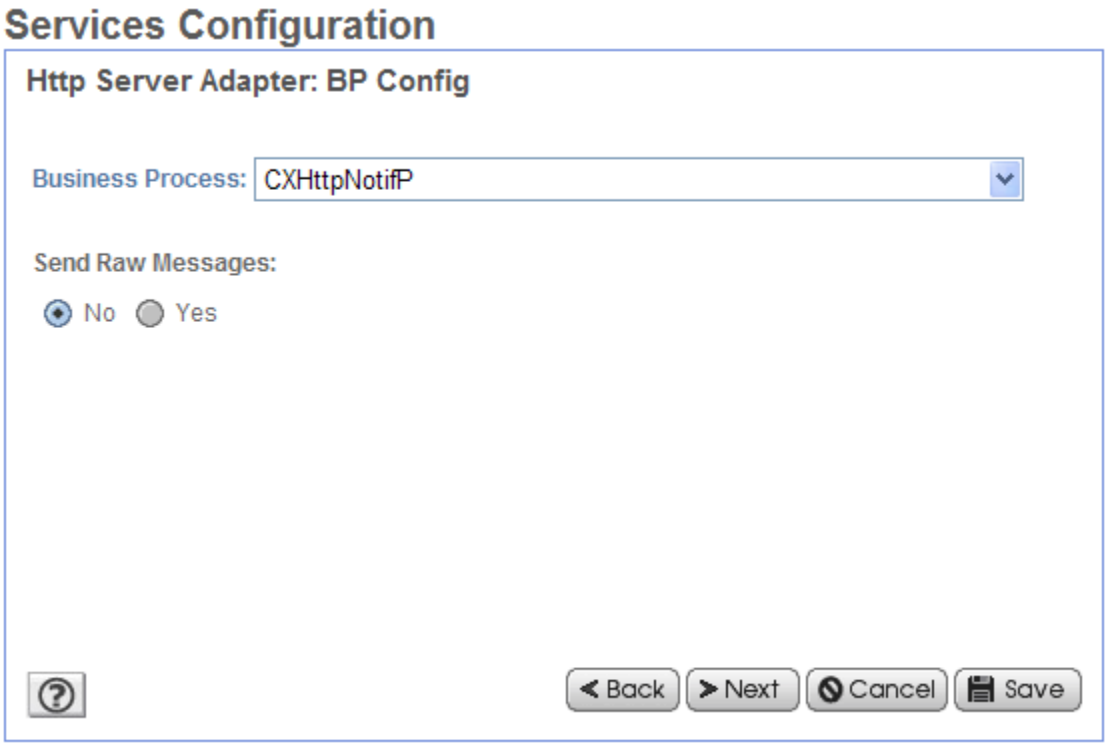


Add the URI /cxnotif. Select « Launch BP ».


Services Configuration
Http Server Adapter: URI
URI
add New URI
edit delete /dashboard
edit delete /DataStore_MsgPrep_Send
edit delete /demosuite
edit delete /DS_Frontend_Import_BP
edit delete /federation
edit delete /hello
edit delete /hello-war
edit delete /myaft
edit delete /onboard
edit delete /portlets
edit delete /soap/SWIFTNetServerRequest
edit delete /soap/SWIFTNetServerSnFRequest
Back
Next
Cancel
Save


Services Configuration
Http Server Adapter: URI: URI Config
URI: /cxnotif
Launch BP Or WAR
Business Process
War File
Back
Next
Cancel
Save

Indicate the name of the business process to launch:



Executing the Business Proccess to Receive the Notification

Executing the Business process described above results in the following:

ProcessData

```
 Process Data

 Process Name: CXHttpNotifP     Instance ID: 289080
 Service Name: CXJournalInst
 <?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIObjectID="localhost:9ed927:111b192d865:-62a9"/>
  <b2b-protocol>http</b2b-protocol>
  <transport-instance-id>HttpServerAdapter_HttpServerAdapter_node1</transport-instance-id>
  <http-request-uri>/cxnotif</http-request-uri>
  <transport-session-id>Mon Apr 02 11:15:31 CEST 2007:8</transport-session-id>
  <output>OK</output>
  <b2b-raw-message>true</b2b-raw-message>
  <HttpRequest>
    <Notification>
      <MonitorIdentification>FEDORA</MonitorIdentification>
      <NotificationType>U02</NotificationType>
      <RequestNumber>09200003</RequestNumber>
      <TransferDirection>T</TransferDirection>
      <PartnerName>BOUCLE</PartnerName>
```

```
      <FileName>FILE01</FileName>
      <PhysicalName>/u1/pga/TOM140/tom1/config/sysin.txt</PhysicalName>
      <Status>E</Status>
      <Trc>0000</Trc>
      <Prc>0000</Prc>
      <Src>0000</Src>
      <Nrc>0000</Nrc>
    </Notification>
  </HttpRequest>
  <CXJournalRequest>
    <Serverid>localhost:9000:X:ADMIN:****:FALSE:0:30:0</Serverid>
    <RequestNumber>09200003</RequestNumber>
    <Mode>GET</Mode>
  </CXJournalRequest>
  <CXJournalGet_4 SCIObjectID="localhost:9ed927:111b192d865:-624b"/>
  <CXJournalResponse>
    <JournalCount>1</JournalCount>
    <ReturnCode>JOURNAL_GET_SUCCESSFUL</ReturnCode>
  </CXJournalResponse>
</ProcessData>
```

Primary Document

```
 PrimaryDocument

 Process Name: CXHttpNotifP     Instance ID: 289080
 Service Name: CXJournalInst
 Document Name: PsHttpDocument_node1_20070402133638974     Document Store:  Database
 Document in process data:   text/xml
  <Notification>
     <MonitorIdentification>FEDORA</MonitorIdentification>
     <NotificationType>U02</NotificationType>
     <RequestNumber>09200003</RequestNumber>
     <TransferDirection>T</TransferDirection>
     <PartnerName>BOUCLE</PartnerName>
     <FileName>FILE01</FileName>
     <PhysicalName>/u1/pga/TOM140/tom1/config/sysin.txt</PhysicalName>
     <Status>E</Status>
     <Trc>0000</Trc>
     <Prc>0000</Prc>
     <Src>0000</Src>
     <Nrc>0000</Nrc>
</Notification>
```

HTTP Notification in Connect:Express Unix

Installing the HTTP Notification

Important Notes: Using HTTP notification with Connect:Express Unix is not compatible with the TCP notification described in « *Connect:Express Unix. Agent_notification* ».
The version of Connect:Express must be 143-1 with Patch 143-106 minimum.

Perform the following operations under the user account of Connect:Express (The installation directory of Connect:Express is defined by the environment variable $TOM_DIR):

1. Stop Connect:Express ($stop_tom)
2. Upgrade the version of Connect:Express if necessary (See Connect:Express documentation)
3. Extract HTTPN_xxx_1.0.3.tar in a temporary directory

```
# tar xvf HTTPN_SOL_1.0.3.tar
```

4. The following files are extracted:

   - tom_httpn (executable)
   - .version
   - httpnotcfg.xml (configuration file)
   - notformat.xml (file defining the form of the notifications)
   - vardef_example.txt (environment variables definition example)

5. Change to the extraction directory and start the installation script:

   The following elements will be necessary:
   a. Installation directory of Connect:Express (referenced by $TOM_DIR).
   b. SI http server IP address or host name (localhost)
   c. SI http server TCP port (10033).

```
# ./install.sh
```

6. Verify that the notification option has been activated in the file $TOM_DIR/config/sysin:

```
HTTPNF=1
```

Note: If a parameter NOTIFY=1 is already defined, change it to 0:

```
NOTIFY=0
```

7. Edit the profile file defining the Connect:Express environment variables and insert the definition of the following new variables (See vardef_example.txt). These variables enable to start, stop and get the status of the notification process tom_httpn.

```
…
# Define these variables in the shell profile of user Connect:Express
export start_httpn=$TOM_DIR/httpn/tom_httpn
export stop_httpn=$TOM_DIR/httpn/"tom_httpn -stop I"
export httpn_status=$TOM_DIR/httpn/"tom_httpn -status"
…
```

8. Execute the profile again and start the monitor ($start_tom)
9. Start tom_httpn process ($start_httpn)

Configuring the HTTP Notification

Tom_httpn process initialization file

The file $TOM_DIR/httpn/httpnotcfg.xml defines the following configuration parameters for the tom_httpn process :

Tag	Description	Défault
url	url of the http server (http://hostname:port/cxnotif or http://ip-address:port/cxnotif)	http://localhost:port/cxnotif
backurl	Optional backup http server url	None
trc	Y/N : Trace of the notifications sent (file tom_httpn.trc)	N
srvtim	Delay in seconds between 2 notification retries	20 (sec)
tim	Time-out in seconds on the tcp/ip network connection with the http server	5 (sec)
env	Y/N : Replace UNIX environment variables in the field PhysicalName of the notification.	Y

Customization

The default notification structure is shown below:

<MonitorIdentification>
<NotificationType>
<RequestNumber>
<TransferDirection>
<PartnerName>
<FileName>
<PhysicalName>
<Status>
<Trc>
<Prc>
<Src>
<Nrc>

To customize the notification edit $TOM_DIR/httpn/notformat.xml and enable/disable elements from the list. You can also change the order of the elements. Refer to the description of CXJournal service for a description of each element.

Using Environment Variables

For Connect:Express Unix, some information depend on the environment variables of the Connect:Express user account. For example PhysicalName can be: $TOM_DIR/in/07100017.tmp. $TOM_DIR is not easy to manage for the SI business process.

Use the <env> option of the $TOM_DIR/httpn/httpnotcfg.xml file to activate/deactivate the substitution of environment variables. The default is Y.

Configuring Transfers for HTTP Notification

To request a notification sent for a transfer, use the notification option in the file definition or the transfer request.

Symbolic File Option (STERM)

The NOT parameter can take one of the following values:

0 to 7:

0: No notification

1: Notification at the beginning of the transfer

2: Notification at the end of the transfer

4: Notification if there is a transfer error

The value x of the parameter is the sum of the values for each kind of notification desired. For example 6 = 2+4 indicates a notification at the end of the transfer or in case of error of transfer.

This is the default value for all transfers of the symbolic file: it can be changed by the transfer request parameter list.

```
C:E UNIX 143-1 ------------------- FILES DIRECTORY ------------------------- tom1
OPTION ===>
SYMBOLIC NAME:              FILE03
INITIALIZATION STATUS .:    E           E:ENABLE H:DISABLE
DIRECTION ..............:   R           T:TRANSMIT R:RECEIVE *:EITHER
RECEIVING PARTNER .....:    $$ALL$$     'NAME',#LISTE, $$ALL$$
TRANSMITTING PARTNER ..:    $$ALL$$     'NAME',#LISTE, $$ALL$$
PRIORITY ..............:    0           0:URGENT 1:FAST 2:NORMAL
DEFINITION TYPE .......:    D           D:DYNAMIC F:FIXED
PRESENTATION TABLE ....:    1           1 -> 9 PRESENTATION TABLE
PARAMETER CARDS FILE:       N           Y/N
SPACE TO RESERVE ......:    N           Y/N
ALLOCATION RULE .......:    0           0:INDIF., 1:PREALL., 2:TO CREATE
PHYSICAL NAME .........:    $TOM_DIR/in/&REQNUMB.tmp
RECORD FORMAT .........:    BU          TF, TV, BF, BU, T*, B*, **
RECORD LENGTH .........:    04000       1-5 NUMERIC CHARAC.
REMOTE DSN (FTP) ......:
TYPE/STRUCTURE/MODE FTP:                E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) ....:                Y/N      FA: N  Y/N   NOT: 2 (0/1/2/3)
OPTION: VIEW                            UPD: 05/10/12 14:59 mverz1
-ENTER- NEXT FIELD                -F3- CANCEL              -F8- COMPLETION
```

Transfer Request (STERM)

The NOT parameter of the transfer request is used in the same way as in the file definition.

```
C:E/UNIX 143-1 -------------- TRANSFER REQUEST ----------------------- tom1
OPTION ===>
FILE ..................: FILE03          DIRECTION .............: T (T/R)
PARTNER ...............: PARTNER.
DPCSID ALIAS ..........: MYNAME..        DPCPSW ALIAS ..........: MYPASSW.
ORIGIN.................: ........        DESTINATION............: ........
SENDER.................:                 RECEIVER ..............:
PHYSICAL NAME .........: $TOM_DIR/out/send.txt.........................
REMOTE PHYSICAL NAME ..: ..............................................
LABEL:.....................................................................
RECORD FORMAT .........: BU               TF, TV, BF, BU
RECORD LENGTH .........: 04000
TYPE/STRUCTURE/MODE FTP: ***              E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) ....: N                Y/N FA: Y/N     NOT: 2 (0 - 7)
TYPE ..................: N                (N/I/H)
TYPE OF CONNECTION ....: T                (X/P/T)
PRIORITY ..............: 0                (0/1/2)
DATE ..................: 19980728101604   (YYYYMMDDHHMMSS)
API FIELD (ETEBAC3: 80 CHARACTERS FOR CARD)
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
DO YOU WANT TO GO ON ?
-ENTER- NEXT FIELD                -F3- CANCEL                  -F8- COMPLETION
```

Batch Transfer Request (p1b8preq)

When you use the batch utility p1b8preq, use subparameter /NTF=x (x = 0 to 7) in the first parameter of the command.

For example:

```
$TOM_DIR/itom/p1b8preq “/SFN=FILE03/DIR=T/SPN=PARTNER/SID=MYNAME/PWD=MYPASSW/NTF=2” \ “/DSN=\
$TOM_DIR/out/send.txt”
```

Using CXTransferSubmit in SI

When using CXTransferSubmit serrvice in a Business process use the TypeOfNotification parameter, which can take values from ‘0’ to ‘7’, and works in the way described above.

Executing tom_httpn Process to Send Notifications

Tom_httpn process executes in the background and sends notifications on demand. Connect:Express writes the notifications in files located in $TOM_DIR/ntfo directory. Tom_httpn reads the notifications and sends them from these files.

Tom_httpn process is started and stopped independently of the monitor.

Commands to Tom_httpn

You can pass the following commands:

Start the Process

```
$start_httpn
```

Immediat Stop of the Process

(preferred method)

```
$stop_httpn
```

Differed Stop of the Process

```
$ tom_httpn –stop D
```

Tom_httpn sends all pending notifications before stopping. The process stops on first sending error. If the number of notifications is large and if the monitor continues writing notifications, it may take a long time before the process stops. Unless you stop the monitor first, you should use the immediate stop.

Displaying tom_httpn Process Status and the Number of Pending Notifications

```
$ tom_httpn -status
```

If tom_httpn is running, the following displays:

```
Status         : STARTED
Message queue id: 196610
Process id     : 2545

Next notification record to write on disk
-----------------------------------------
Record number: 324
Current file: N20070421174034

Next notification record to send to http server
-----------------------------------------------
Record number = 132
Current file  = N20070421174034

Curent list of notification files
---------------------------------
N20070421174034
```

```
  Current number of notifications waiting to be sent: 292
```

If tom_httpn is not running, the following displays:

```
  Status         : NOT STARTED
  Message queue id: -
  Process id     : -

  Next notification record to write on disk
  -----------------------------------------
  Record number: 324
  Current file: N20070421174034

  Next notification record to send to http server
  -----------------------------------------------
  Record number = 132
  Current file  = N20070421174034

  Curent list of notification files
  ---------------------------------
  N20070421174034
  Current number of notifications waiting to be sent: 292
```

Activating/Deactivating the Trace

```
$ tom_httpn –trc Y/N
```

The trace is written in a text file: $TOM_DIR/httpn/tom_httpn.trc.

Miscellaneous commands

The following commands are valid only if tom_httpn is stopped:

Send Notification by Batch Processing

```
$ tom_httpn –wait N
```

Tom_httpn starts, sends all pending notifications and stops.

Purge Pending Notifications

```
$ tom_httpn –purge all
$ tom_httpn –purge HH:MM
$ tom_httpn –purge AAAA/MM:JJHH:MM:SS
```

Example: purge all pending notifications older than today 10:00 AM:

```
$ tom_httpn –purge 10:00
```

Backup http server

It is possible to redirect automatically the notifications to a backup http server, when the primary http server is stopped. In this case, define a backup url in the configuration file httpnotcfg.xml, as shown in the following example:

```
<trc>S</trc>
<tim>5</tim>
<srvtim>20</srvtim>
<env>Y</env>
<url>http://10.87.15.35:10033/cxnotif</url>
<backurl>http://10.87.15.34:10033/cxnotif</backurl>
```

The command $httpn_status displays the active url to which the notifications are currently been sent.

```
  Status                : STARTED        Message queue id : 19857408
  Process id            : 15074
  url (inactive)        : http://10.87.15.35:10033/cxnotif
  backup (active)       : http://10.87.15.34:10033/cxnotif
  retry timer (srvtim)  : 20 seconds     connection timer (tim): 5 seconds
  trc                   : Short                           env : 1

  Next notification record to write on disk
  -----------------------------------------
  Record number : 31
  Current file  : N20081024102139

  Next notification record to send to http server
  -----------------------------------------------
  Record number = 29
  Current file  = N20081024102139

  Current list of notification files
  ----------------------------------
  N20081024102139
  Current number of notifications waiting to be sent: 2
```

If both servers are stopped, tom_httpn tries alternately to send the notifications to each url until one server has been restarted.
After automatic redirection to the backup url and when the primary http server is restarted, it is necessary to use a manual command to redirect again the notifications back to the primary server:

```
tom_httpn –activate primary
```

The activation command has the following general form:

```
tom_httpn –activate primary|backup
```

It is also possible to lock the transmission of the notification to one specific url by using the command:

```
tom_httpn –lock primary|backup
```

In this case, the automatic backup is deactivated.

Managing Monitor, SI or Httpn Stop.

All three components are running independently: no notification can be lost.

If tom_httpn stops, the monitor writes the notifications in files placed in the ntfo directory. When tom_httpn initializes, all pending notifications are sent to SI.

If the monitor is stopped, tom_httpn sends all pending notifications and returns in wait state.

If SI is stopped, tom_httpn retries periodically to send the notifications to SI. The retry interval is set in the <srvtim>20</srvtim> parameter of httpnotcfg.xml.

Notifications Sequence

Notifications are sent in a FIFO sequence. No notification is sent before all previous notifications have been sent.

The only way to cancel notifications is to stop tom_httpn, use the purge command, and start tom_httpn again.

Should any network error occur while sending a notification to SI, tom_httpn retries periodically to send it every 20 seconds (srvtim parameter), until the notification is successfully sent.

Troubleshooting

Use commands *tom_httpn –status* and *ps* to check the status of the http notification system.

No Notification created by the Monitor

1. Check that the HTTPNF parameter is set to 1 in $TOM_DIR/config/sysin
2. Check via STERM that the notification was requested for the transfer request. If it was not requested, check the symbolic file definition and/or the request parameters.

No Notification sent to SI

1. Check that tom_httpn is active, using ***tom_httpn –status*** and ***ps –ef | grep tom_httpn***
2. If tom_httpn is active, activate the trace with the command **tom_httpn –trc Y**. The trace is written in $TOM_DIR/httpn/tom_httpn.trc: analyze the trace and search for any network problem.

tom_httpn Does not Start

1. Check that tom_httpn is not already active using **tom_httpn –status** and ***ps –ef | grep tom_httpn***
2. If the status is « active » although ***ps*** does not show tom_httpn process, tom_httpn might have been abnormally stopped (kill -9 or core dump for example). In this case you must purge the IPC queue that has been created during previous execution. Use tom_httpn –status to retrieve the identifier of the message queue. Use the Unix command: ***ipcrm –q <identifier>***. Then remove all file pid.<process-id> from the $TOM_DIR/httpn directory, if any.
3. Another reason for tom_httpn not initializing is an invalid httpnotcfg.xml file.

Purging the Notification System

1. Stop the monitor.
2. Remove the $TOM_DIR/ntfo directory and all files.
3. Remove the trace file $TOM_DIR/httpn/tom_httpn.trc if any
4. Start the monitor

HTTP Notification in Connect:Express Windows

HTTP notifications generated by the monitor are sent to the remote SI by a specific executable tom_httpn. The monitor, tom_httpn and SI can be stopped or restarted independently, without loss of notifications.

Note: With Connect:Express Windows, the http notification can be used together with the TCP notification described in « *Connect:Express Windows. User Guide* ».
Connect:Express Windows version must be V3.0.5.001 minimum.

Installing the HTTP Notification

The HTTP notification component is not installed at Connect:Express monitor installation. It requires an additional installation on a beforehand installed monitor.

Perform the following operations:

1. Stop Connect:Express
2. Upgrade the version of Connect:Express if necessary (See Connect:Express documentation)
3. Copy HTTPN_WIN_1.0.3.zip to an empty temporary folder, for example c:\tmpdir.
4. Extract the archive (with WinZip for example) in this directory.
5. Execute setup.exe. Indicate during the installation the root directory of the monitor.

Uninstalling the HTTP Notification

Perform the following operations:

1. Select « Start / Settings / Control Panel / Add or Remove Programs » from the Windows task bar.
2. Select « Connect:Express HTTP Notifications » in the list of installed programs, the click « Remove ».

Configuring the HTTP Notification

You can manage the notification structure.

Customization

The default notification structure is shown below:

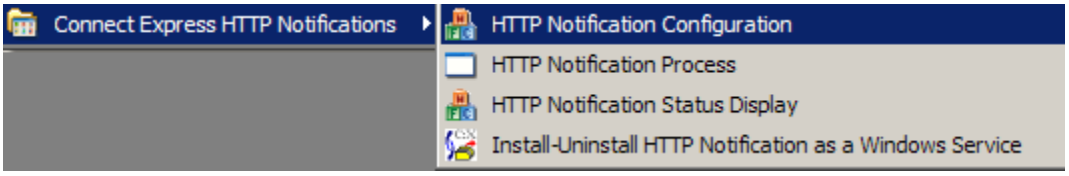
```
<MonitorIdentification>
```

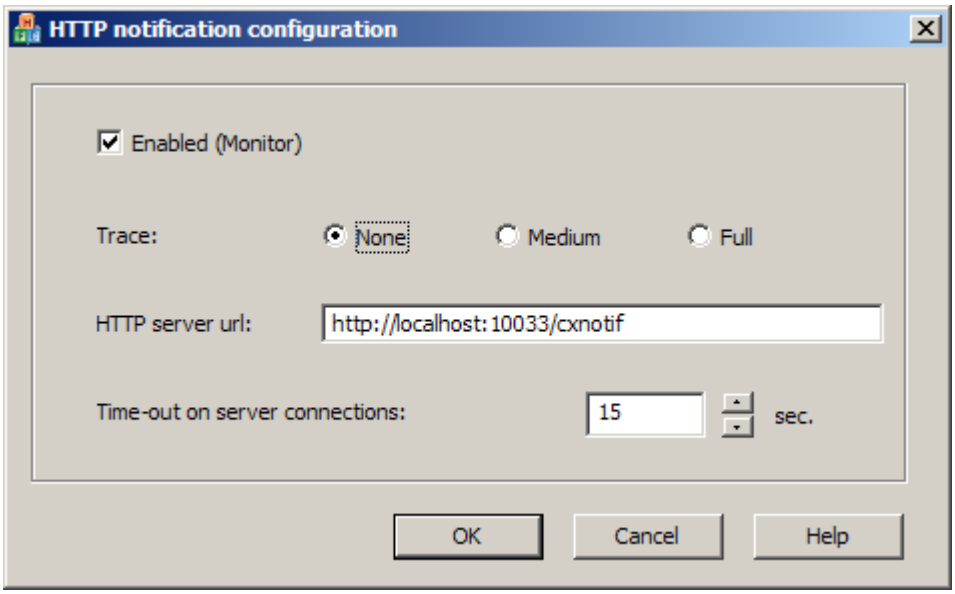
<NotificationType>
<RequestNumber>
<TransferDirection>
<PartnerName>
<FileName>
<PhysicalName>
<Status>
<Trc>
<Prc>
<Src>
<Nrc>

To customize the notification edit the file **notformat.xml** located in the Connect:Express directory and enable/disable elements from the list. You can also change the order of the elements. Refer to the description of CXJournal service for a description of each element.

Httpn_config.exe Utility used to configure the tom_httpn Process

This utility can be started from the menu « Start / Programs »:





At installation, HTTP notifications are configured with the above settings.

The configurable settings are:

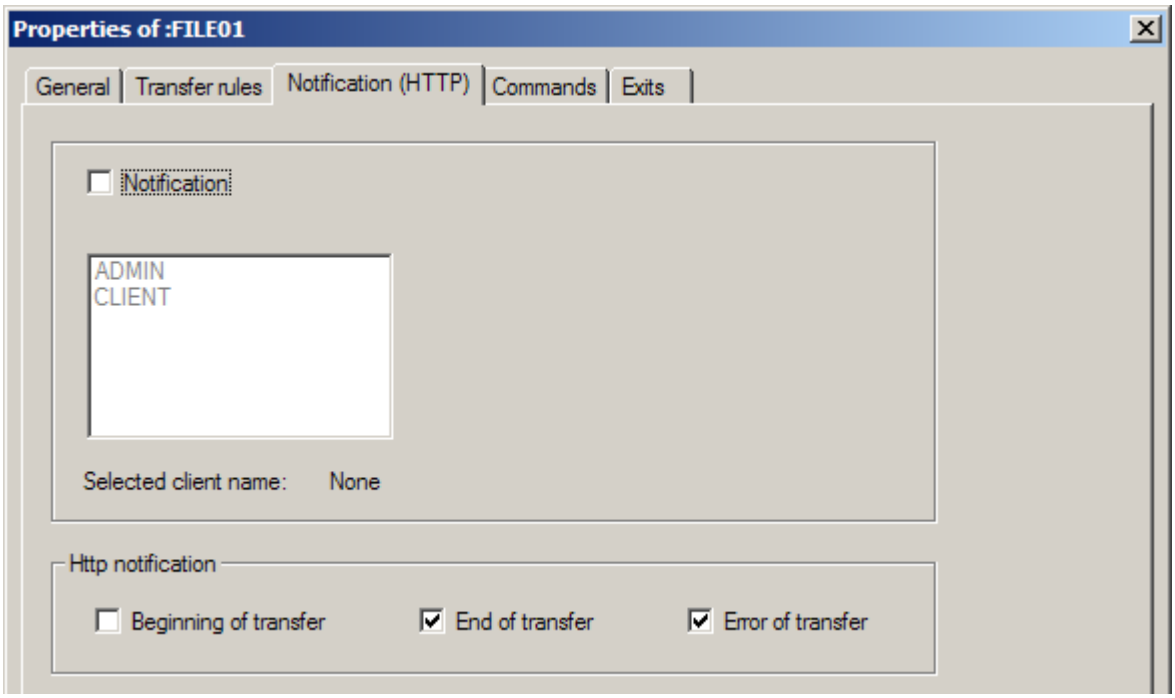
Enabled (Monitor):
Indicates if the monitor is enabled to create HTTP notifications at the beginning, end or error of transfers.
If checked, the monitor creates HTTP notifications that can be sent to the HTTP server by tom_httpn. A stop-restart of the monitor is necessary for this parameter to be effective.
Trace:
Trace level of the process tom_httpn. The trace file is trace\tom_httpn.txt.
HTTP server url:
Url of the HTTP server, to which the notifications are sent.
Time-out on server connections:
Time-out value in seconds for the connections to the server.

Configuring Transfers for HTTP Notification

To request a notification to be sent for a transfer, use the notification options in the symbolic file definition or in the transfer request. Setting these options in the symbolic file definition determines the notification conditions for transfers in server mode (For example server/receiver mode).
Setting these options in the parameters of a transfer request determines the notification conditions for transfers in requestor mode. (For example requestor/sender mode).

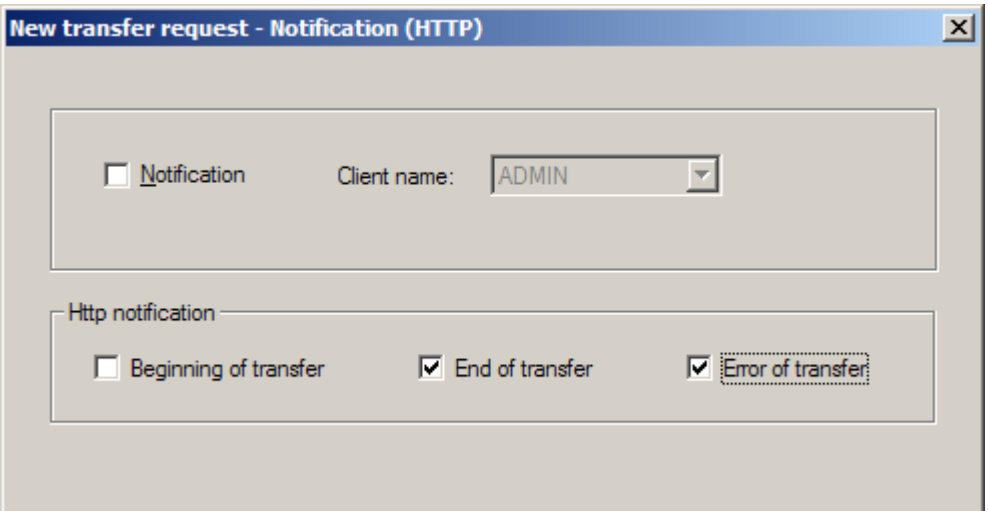
Symbolic File

The http notification options for a symbolic file are presented by the monitor's graphical interface as below:



Transfer request (Graphical interface iu_tom)

When requesting a transfer with the graphical interface, the http notification options are indicated by the following dialog of the graphical interface:



Batch Transfer Request (Tomreq.exe)

When you use the batch utility Tomreq.exe, use parameter /L:x (x = 0 to 7).

0: No notification

1: Notification at the beginning of the transfer

2: Notification at the end of the transfer

4: Notification if there is a transfer error

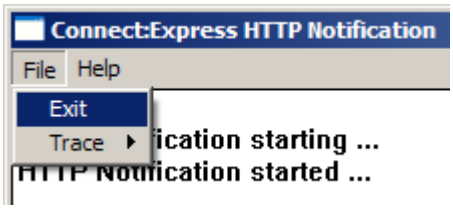
The value x of the parameter is the sum of the values for each kind of notification desired. For example 6 = 2+4 indicates a notification at the end of the transfer or in case of error of transfer.

```
"c:\cexpress\Tomreq.exe" /L:6 /F:FILE01 /P:BOUCLE /S:T /T:N /D:"c:\cexpress\out\out.txt" /C:ADMIN /M:ADMIN
/K:I /H:localhost /O:7000 /G:"c:\CExpress\Tomnt.ini" /A:BOUCLE /W:PSW
```

Starting/Stopping the tom_httpn Process

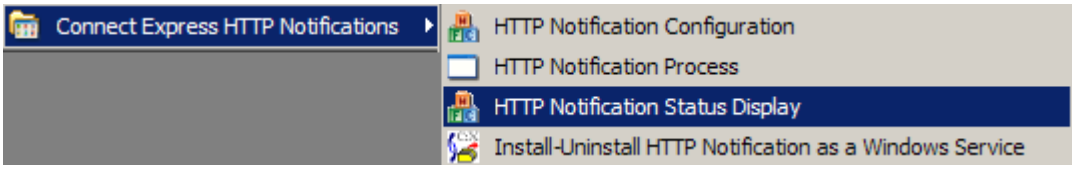
The process tom_httpn can be started from the menu « Start / Programs »:

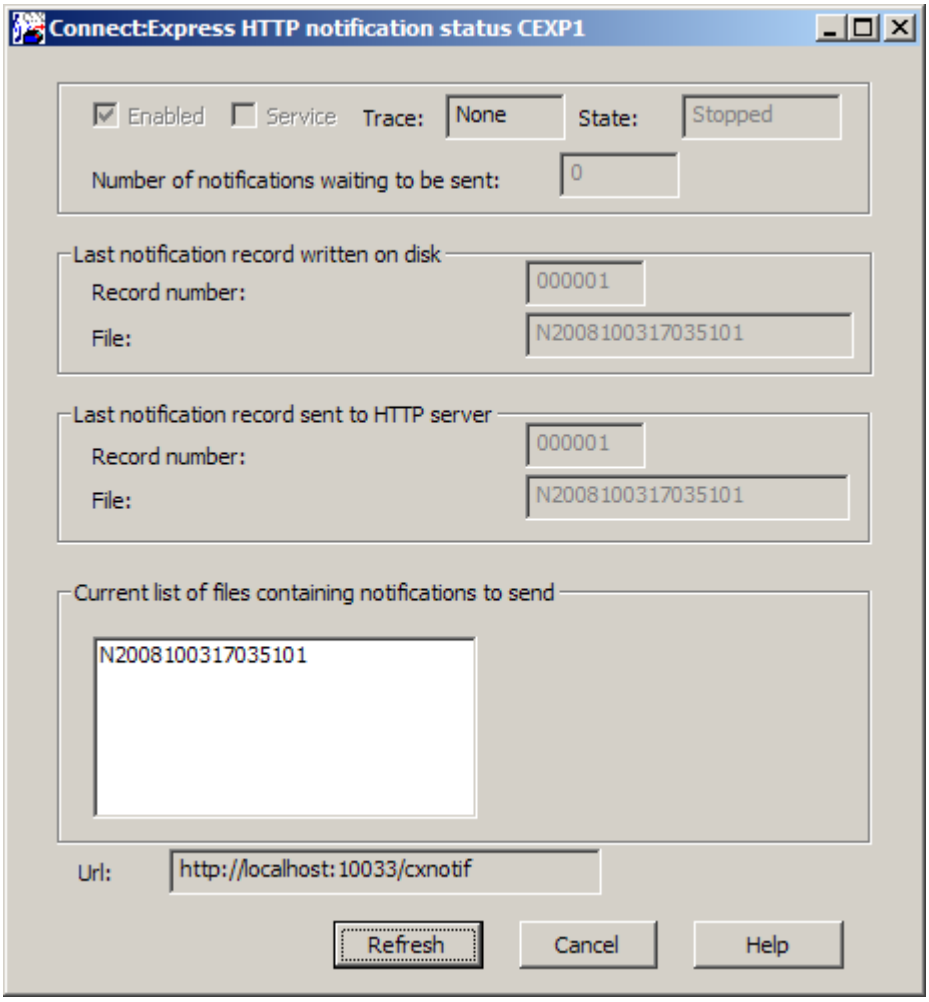
To stop, use the exit menu of the tom_httpn windows.



Httpn_status.exe Utility Showing Notifications Waiting to be Sent

This utility can be started from the menu « Start / Programs »:

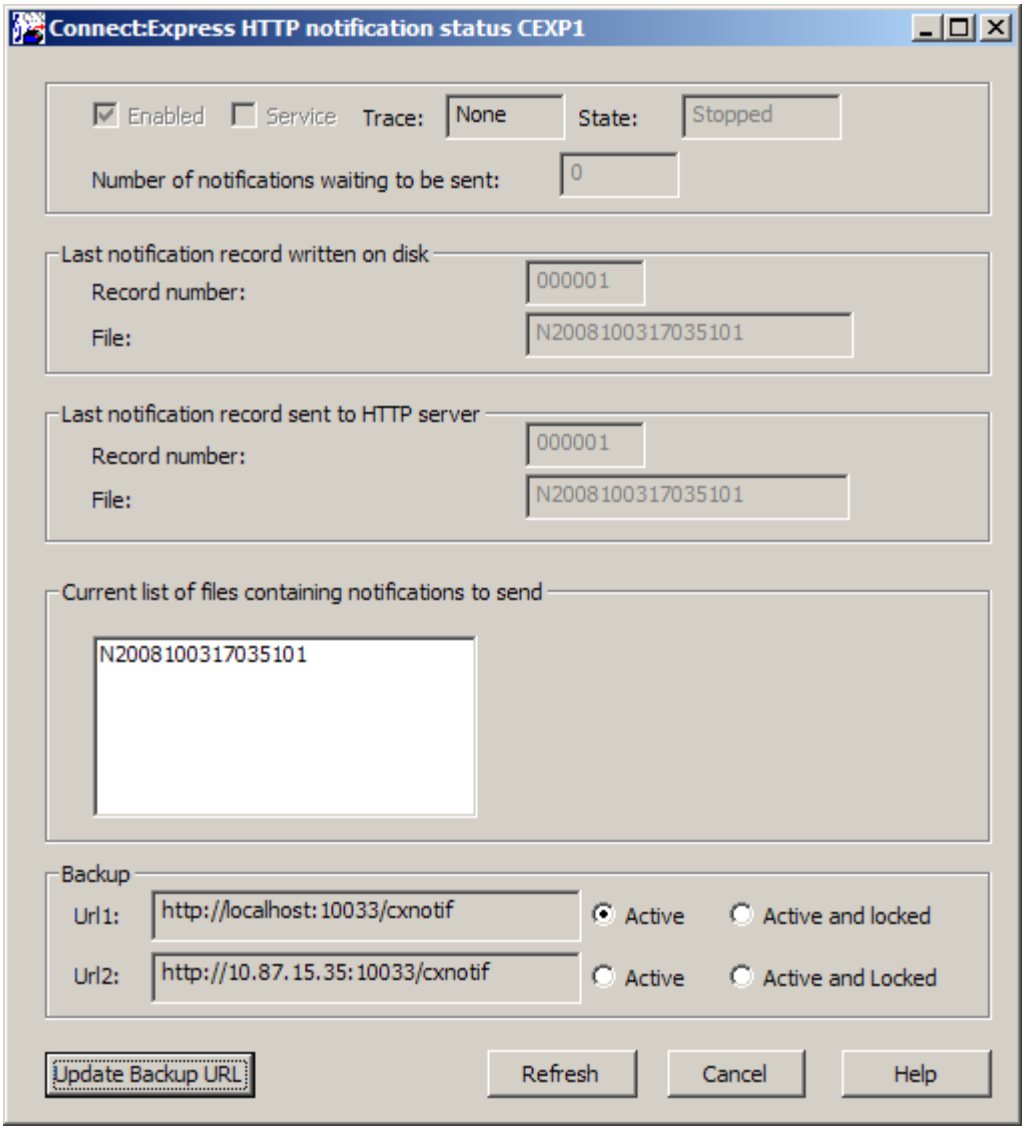




Enabled:
Indicates if the monitor is enabled to create HTTP notifications at the beginning, end or error of transfers.
If checked, the monitor creates HTTP notifications, which can be sent to the HTTP server by tom_httpn.
Service:
Indicates if the process tom_httpn is installed as Windows service.
Trace:
Indicates if the trace of tom_httpn is active. The trace file is trace\tom_httpn.txt
State:
State (Started or stopped) of the tom_httpn process.
Number of notifications waiting to be sent:
Indicates the number of notifications created by the monitor and waiting to be sent by tom_httpn.
Last notification record written on disk:
Record number:
Record number, in the file below, of the last notification written by the monitor.
File:
Name of the notification file created by the monitor in the ntfo directory. Each notification file contains up to 1000 notifications written sequentially by the monitor.
Last notification record sent to HTTP server:
Record number: Record number of the last notification sent to the remote HTTP server by tom_httpn.
File:
Notification files are read by tom_httpn and purged after complete emission of all notifications.

Current list of files containing notifications to send:
List of files containing notifications waiting to be proccessed by tom_httpn.
Url:
HTTP server url.

If a backup URL has been configured, httpn_status displays the following dialog box:



It is possible to manually activate or lock the transmission of the notifications to one of both servers, by clicking on the appropriate radio button and updating with the “Update Backup URL” button.

For example :

If S1 and S2 are the 2 HTTP servers, if S1 is down, the notifications are automatically redirected to S2. Even if S1 restarts, they keep on being sent to S2 while S2 is up. To redirect again towards S1, a manual operation is necessary. Set in this case the “Active” button for Url1. On the other hand, if S2 is stopped, automatic backup towards S1 is done.

The “Active and locked” buttons enable to lock the transmission of the notifications to a specific server. In this case, the backup is deactivated.

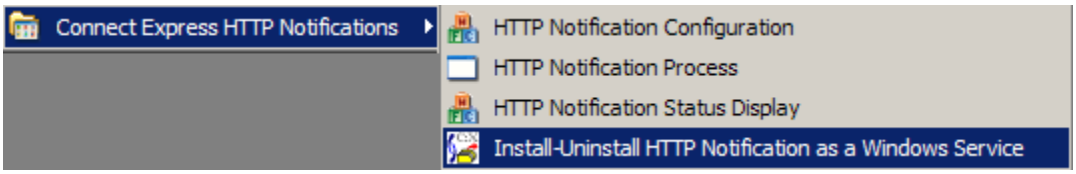
Httpn_cmd.exe utility

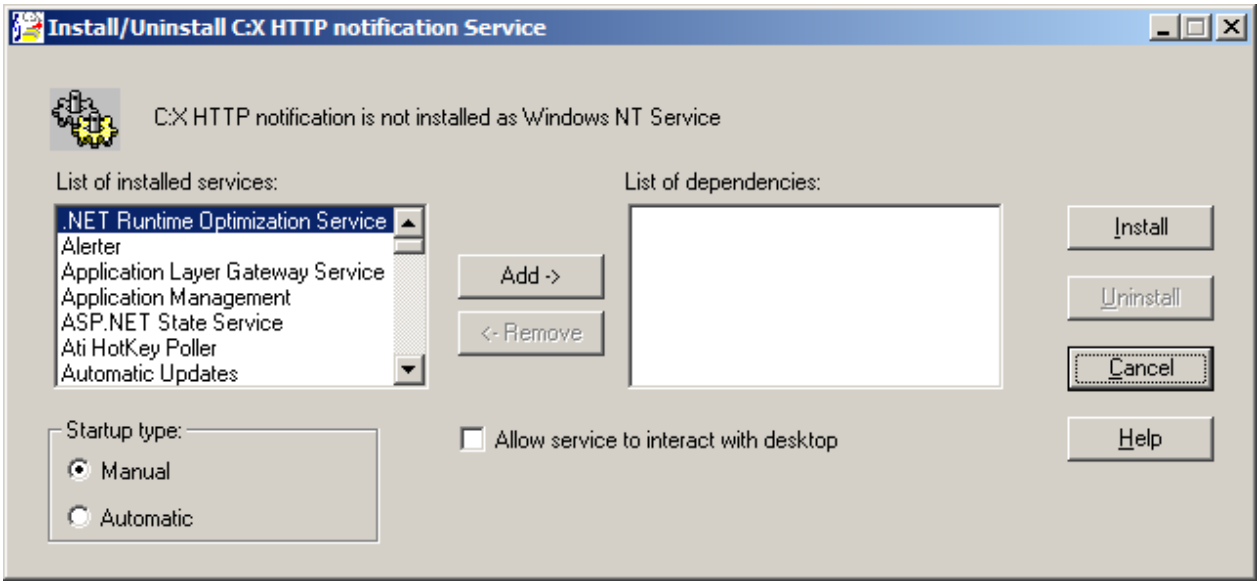
This utility is a batch utility enabling to stop the process tom_httpn, to activate or deactivate the trace and to update the backup flags.

```
Usage: httpn_cmd stop
Usage: httpn_cmd trace [medium|full|off]
Usage: httpn_cmd [activate|lock] [1|2]
```

Httpn_service.exe Utility for tom_httpn Installation / Uninstallation as Windows Service

This utility can be started from the menu « Start / Programs »:





This utility is used to install or uninstall the Connect:Express HTTP notification process as a Windows service.

The installation as a Windows service ensures that the HTTP notification functionality is permanently available in different ways:

- Automatically started at system startup
- Associated with other dependent services that must be started before the C:X Service
- Locally or remotely stopped/started using the Windows Service Manager
- Locally or Remotely supervised using the Windows Events Viewer
- Executed under system or user account with the associated rights

Purging the Notification System

5. Stop the monitor.
6. Delete all files in the ntfo directory of Connect:Express.
7. Start the monitor

Notification Fields

The table below provides the list of notifcation fields, in alphabetic order, with length. It shows that some fields ar not available on all platforms.
Le field *NotificationType* identifies the platform, the type of notification and the version of the notifcation. The type of notification is common to the three platforms (I = Initialization, E =End, R = Error). The values of the *Status* field, on the contrary, may be platform dependent (Status=E is common to z/OS, Unix and Windows).

Field	Lg max	Description	u	w	z
AppcModeName	8	Sna Lu6.2 ModeName		x	x
AppcTpName	8	Sna Lu6.2 Tp Name		x	x
ClientName	8	Name of the client (or administrator) to notify	x	x	x
Erc	4	Ctree return code		x	
ExternalRequestNumber	8	Request number on the remote side (Type Tom only, PeSIT)	x	x	x
FileBytes	12	Number of bytes of the file	x	x	x
FileName	8	Symbolic file name	x	x	x
FileNumberOfRecords	12	Number of records sent/received	x	x	x
FileOpenOption	1	Allocation rule, N = New file, R = Replace, O = Pre-allocated, U = User	x	x	x
FileOrganization	1	File organization, S = Sequential, I = Indexed, R = Relative, P = Pds, V = Vsam	x	x	x
FileRecordLength	5	Local record length	x	x	x
Label	80	File user identification or label (Pi37 PeSIT)	x	x	x
LocalName	8	Alias name of the local Connect:Express	x	x	x
MonitorIdentification	8	Name of the Connect:Express server	x	x	x
NetworkBytes	12	Number of bytes transferred	x	x	x
NetworkMessageSize	5	Network message size	x	x	x
NotificationLink	1	Type of link used for sending the notification: T = TcpIp	x	x	x
NotificationType	3	Os,Type,Version: Os = Z/os,Unix,Windows Type = Initialization,End,Reject	x	x	x

		Version = 2 Example: WE2 = Windows, End of Transfer, Version 2			
Nrc	12	Network Return code	x	x	x
OriginalPhysicalName	44	File name proposed to remote as their data set name (or Pi99)	x	x	x
PartnerName	8	Symbolic Partner name	x	x	x
PhysicalName	127	Local file physical name	x	x	x
Pi61	24	The entity that is processing the file before sending it (PeSIT)	x	x	x
Pi62	24	The entity that is processing the file after receiving it (PeSIT)	x	x	x
Prc	3	Protocol return code	x	x	x
Priority	1	Transfer priority, 0 = Urgent, 1 = Fast , 2 = Normal	x	x	x
Protocol	1	Transfer protocol, D=PesitD, E=PesitE, O=Oftp, 3=Etebac3, F=ftp	x	x	x
RealCompression	4	Compression performed (PeSIT, Odette)	x	x	
RegistrationDate	10	Date and time when the notification record is created by Connect:Express	x	x	x
RegistrationTime	8		x	x	x
RemotePhysicalName	44	Remote file physical name (Type Tom only, PeSIT)	x	x	x
RequestDate	10	Date when the request is accepted by Connect:Express	x	x	x
RequestTime	8	Time when the request is accepted by Connect:Express	x	x	x
RequestNumber	12	Request number given by Connect:Express	x	x	x
Requestor	8	The name of the entity (user, job ..) that submitted the request	x	x	x
RequestStatus	1	Status of the request after transfer: purged (Y) or not (N)	x	x	x
RetryNumber	2	Number of retries for the request	x	x	x
SendReceiveCount	12	Number of network messages moved	x	x	x
SnaLuName	8	Sna LU name		x	x
SnaRc1	4	Sna return code		x	
SnaRc2	8	Sna return code		x	
Src	8	System Return code	x	x	x
Status	1	Transfer status – this field is platform dependent **z/OS**: C = In progress / E = Ended / S = Selection Error / I = Interrupted / W = Connection Error / Q = Inbound request in progress **Windows**: A or W = Active / C = In progress / E = Ended / S = Selection Error / I = Interrupted **Unix**: C = In progress, O = Selection Error or Interrupted, E = Ended	x	x	x
TcpIpAddress	15	Remote TCP/IP address	x	x	x
TcpIpHost	127	Remote TCP/IP host name	x	x	x
TcpIpPort	5	Remote listening TCP/IP port	x	x	x
TcpipRc	4	TCP/IP return code	x	x	x
TransferBeginningDate	10	Beginning of transfer date and time	x	x	x
TransferBeginningTime	8		x	x	x
TransferDestination	8	The entity that is receiving the transfer request (Pi4bis PeSIT)	x	x	x
TransferDirection	1	Transmission or Reception	x	x	x
TransferEndDate	10	End of transfer date and time	x	x	x
TransferEndTime	8		x	x	x
TransferIdentifier	8	Transfer identifier exchanged with the partner (PeSIT)	x	x	x
TransferOrigin	8	The entity that is initiating the transfer (Pi3bis PeSIT)	x	x	x
Trc	4	Connect:Express Return code	x	x	x
TypeOfData	1	Type of data, A = Ascii, E = Ebcdic, B = Binary	x	x	x
TypeOfFile	2	Type of file, TF = Text fixed, TV = text variable, UF = Unix fixed, UV = Unix Variable, BF = binary fixed, BU = Binary undefined S = Sequential, V = VSAM, P = PDS, PE = PDSE, PU = PDS unload, VU = VSAM unload, SU = SYSOUT unload, UU = User unload, H = HFS, TU = Sequentiel+Edciconv, HU = HFS+Edciconv	x	x	x
TypeOfLink	1	Type of link, 0 = LU 6.2, 1 = X25, 2 = TCP/IP, M=mixed	x	x	x
TypeOfPartner	1	Type of Partner, Other or Tom	x	x	x
TypeOfRequest	1	Type of request, N = Normal, I = Inquiry, H = Hold	x	x	x
TypeOfUser	1	Type of user, I = Internal, E = External	x	x	x
UserDataReceived	254	User information received with the file (Pi99 PeSIT)	x	x	x

UserDataSent	254	User information sent with the file (Pi99 PeSIT)	x	x	x
UserRequestID	16	Identification of the request given by the user		x	
X25Cause	2	X25 Cause	x	x	
X25Diag	2	X25 Diagnostic	x	x	
X25Facilities	32	Remote X25 address, facilities	x	x	x
X25LocalAddress	15	Local X25 address	x	x	x
X25Localport	1	Local device	x	x	
X25MchLuName	8	Local Mch name			x
X25MchName	8	Local Mch Lu name			x
X25Rc	4	X25 Return code	x	x	
X25RemoteAddress	15	Remote X25 address	x	x	x
X25UserDataField	16	Remote X25 address, user data field	x	x	x

This chapter describes how to implement a Connect:Express Browser Interface under control of the Http server of SI. It describes first the installation procedure and the first tests that you can perform. In a third paragraphe it shows how to integrate this Human interface in SI environment.

Connect:Express User Interface

A browser interface is provided for integration of Connect:Express file transfer functionnality with SI. This interface takes advantage of the http server of SI : it can be integrated in an application by the use of an URI configured in the http server adapter.

This interactive facility complements the set of services described above, that you can use for automatizing of the file transfer process managed by Connect:Express, PeSIT for example. Through screens you can perform the average operations related to file transfer with Connect:Express : partners and files management, transfer request and monitoring.

Installing the Interface

First of all extract on your desktop the two files of the CXSI_IHM_5.0.05.zip file: CXSession.war and CX_IHM_Export.xml.

Deploying the Interface

Copy the .war file in SI environment: [*installdir*]/noapp/deploy. SI wil deploy it during next initialization.

Installing the SI Ressources

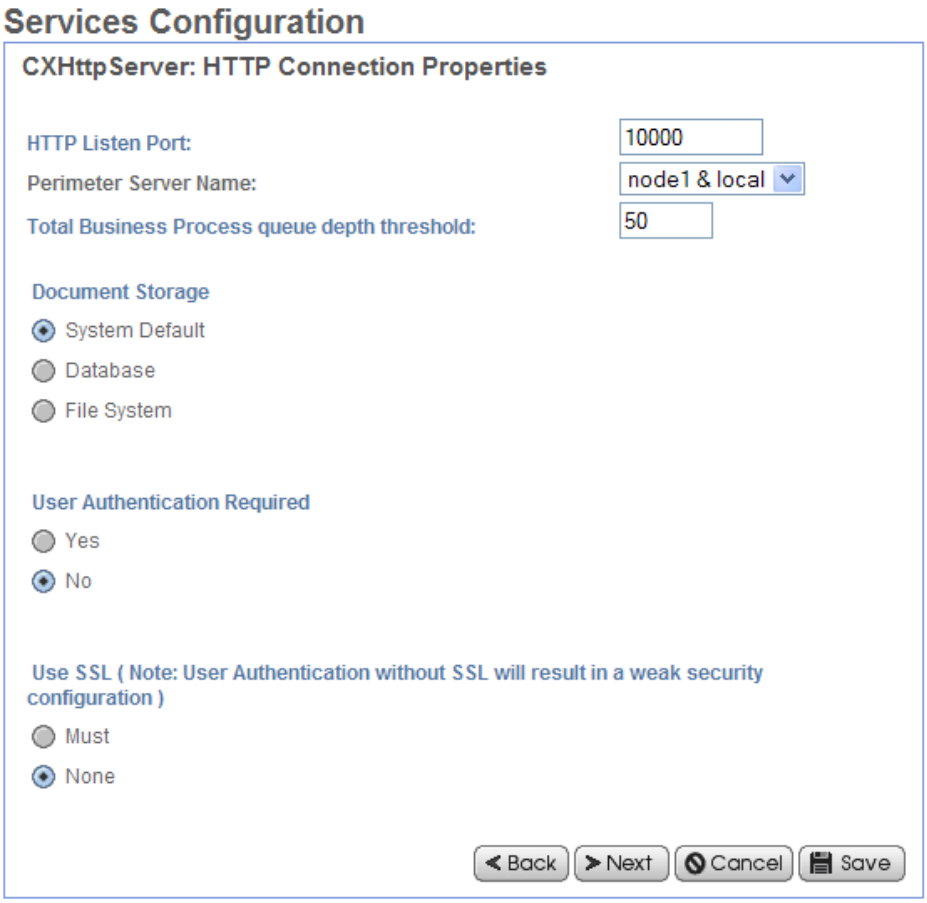
The .xml file includes a BP 'CXConfig', a configuration CXConfig of CXConfigurationAdapter, a configuration CXHttpserver of HttpServerAdapter .

Login to the SI server and import the file CX_IHM_Export.xml using the menu deployment / resource manager / import. Provide the following parameters:

- File Name *CX_IHM_Export.xml*
- Passphrase *connect:express*
- *Skip Tag Name and Tag Description*
- Update Objects *Yes*
- BP Selection select CXConfig
- Service Configurations select both services

The confirmation screen displays :

Configure CXHttpServer using the menu *deployment / services / configuration* : update *Http Listen Port* field and the *War File Pass* field of the URI CXSession.



Services Configuration

Services Configuration

The confirmation screen displays :

Description of the Interface

Before you make tests with the interface you must restart SI after the intallation procedure and start the monitor.

Connect to the URI as shown below. The parameter *cxostype* can take two values, WINDOWS or UNIX according to the platform were the monitor executes.

http://*serveursi*:10000/CXSession?cxipaddr=xx.xxx.xxx.xxx&cxipport=nnnnn&cxostype=WINDOWS

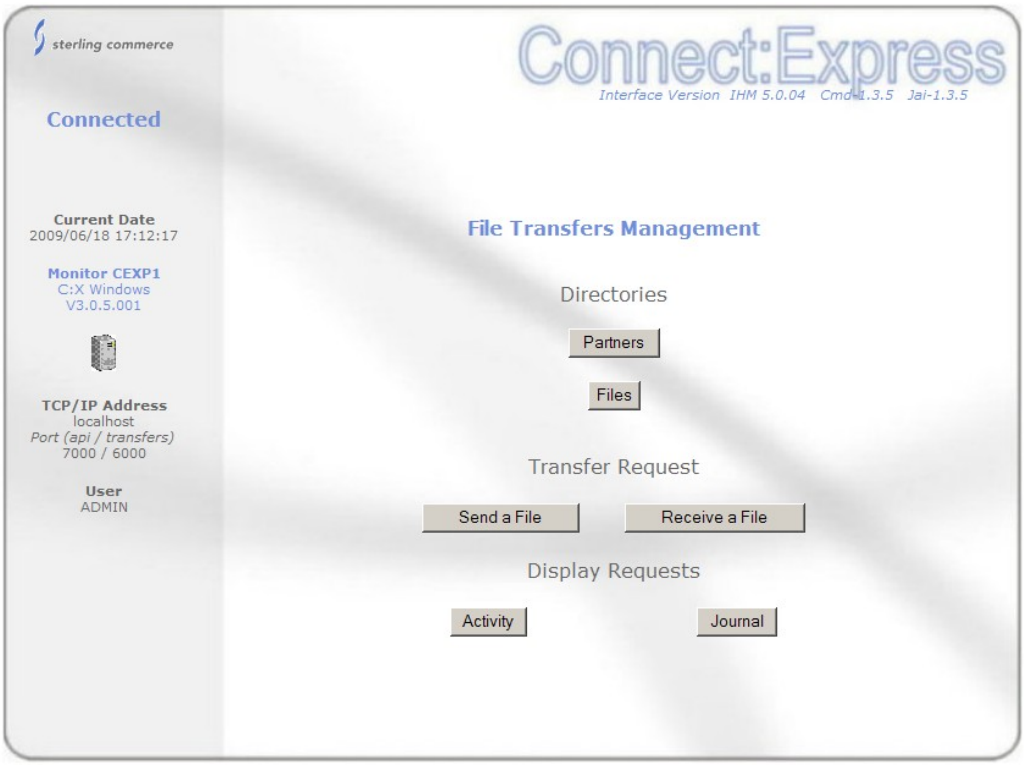
Remark : if you connect to URI http://*serveursi*:10000/CXSession, the following default values are taken:

- cxipaddr=localhost
- cxipport=7000
- cxostype=WINDOWS

In case of a WINDOWS monitor the defaults user name and password are *ADMIN* and *ADMIN*.

This section describes some specificities of the browser interface. Refer to Connect:Express documentation for any information regarding the implementation of file transfers.

The menu displays.



On the left side some current information is shown about the work and the monitor. The workbench is on the left side.

The name of the monitor, CEXP1, is shown. This is the DPCSID field for C:X Unix, or the name that you gave to C:X Windows during the installation procedure.

The versions of the interface, of the CXCMD and the CXJAI components are displayed respectively on the top left corner of the workbench. This enable you to check the compatibility of all components.

If you place the mouse on the icon representing a computer, you can see the operating system on which the monitor executes.

You can pass the language as parameter to the URI. The following example shows the menu for a Unix monitor. The screens are in French.

http://si:10000/CXSession/?cxipaddr=199.1.4.53&cxipport=5050&cxostype=UNIX&lang=FR

In such configuration, all functions are based on the Connect:Express API, with no SI Business Process involved.

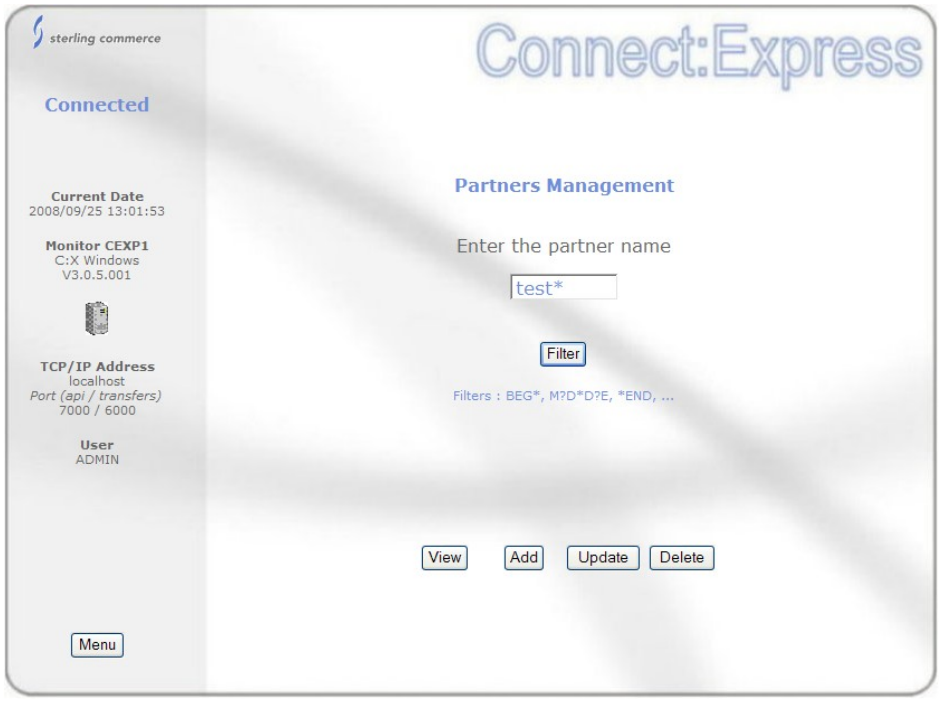
Managing Directories

You can create, update and delete Partners and Files. The interface provides a filter : you can either use the filter or fill in the name directly. The field is required, letters are set to uppercase automatically.

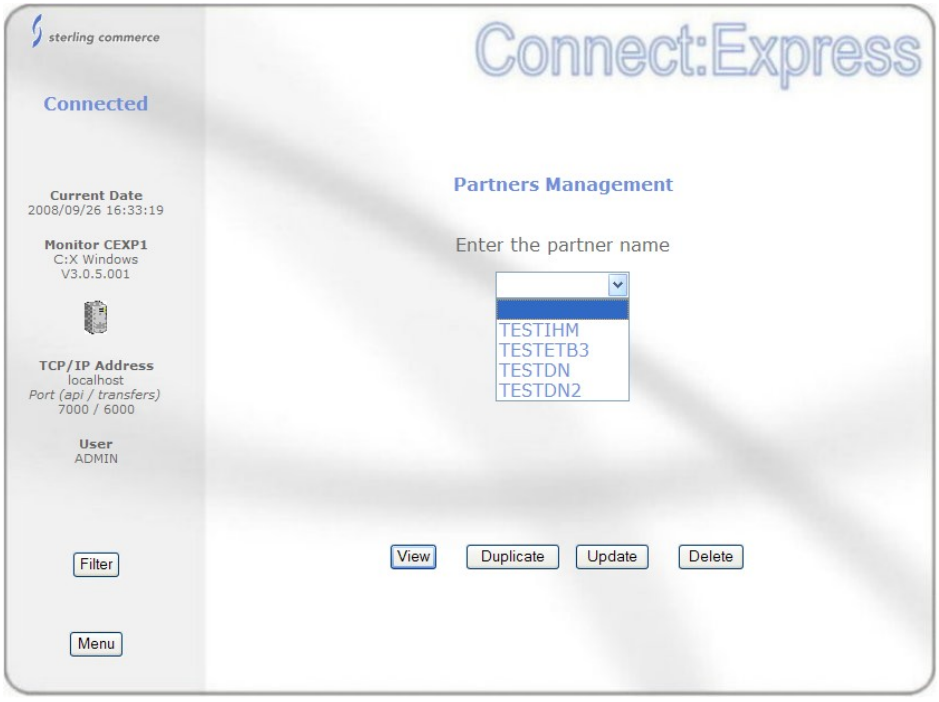
In filter mode, The field supports wild characters '*' and '?'.

Examples :

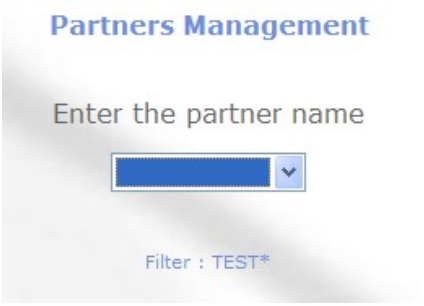
- '*' = all names
- 'RAC*' or 'RAC' = all names starting with RAC
- 'RAC ?0' = all names starting with 5 characters : 'RAC', any character, then character '0'.



Click filter.

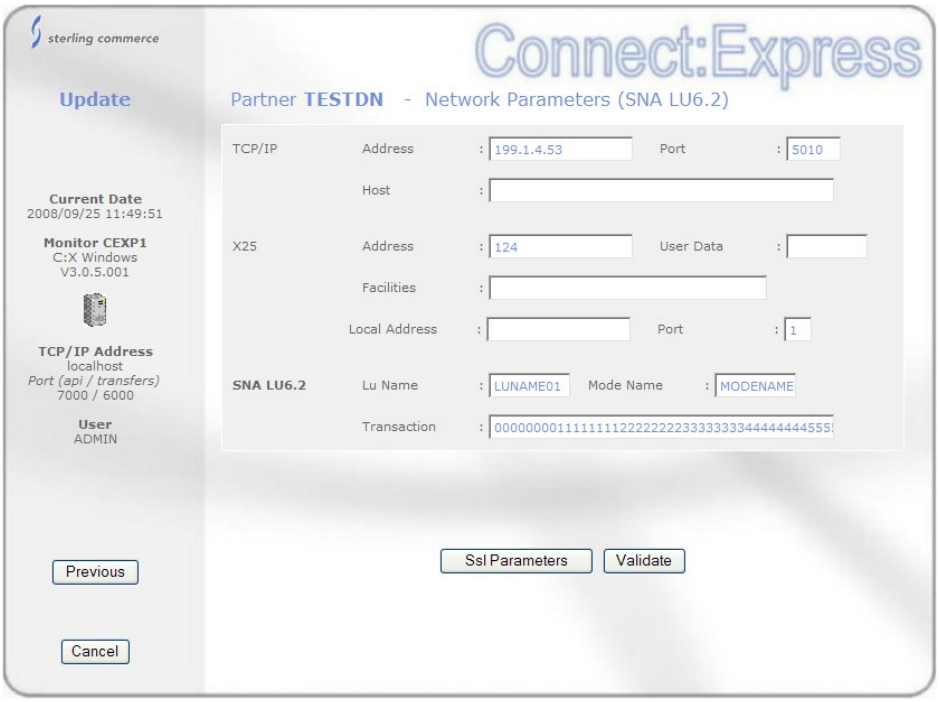


Using filter enable you to duplicate a definition. The active filter is shown under the input field. Use the *Filter* button on the left side to change filter.



Screens are structured as the following : buttons on the right part enable you to go forward, buttons on the left side enable you to go backwrad :

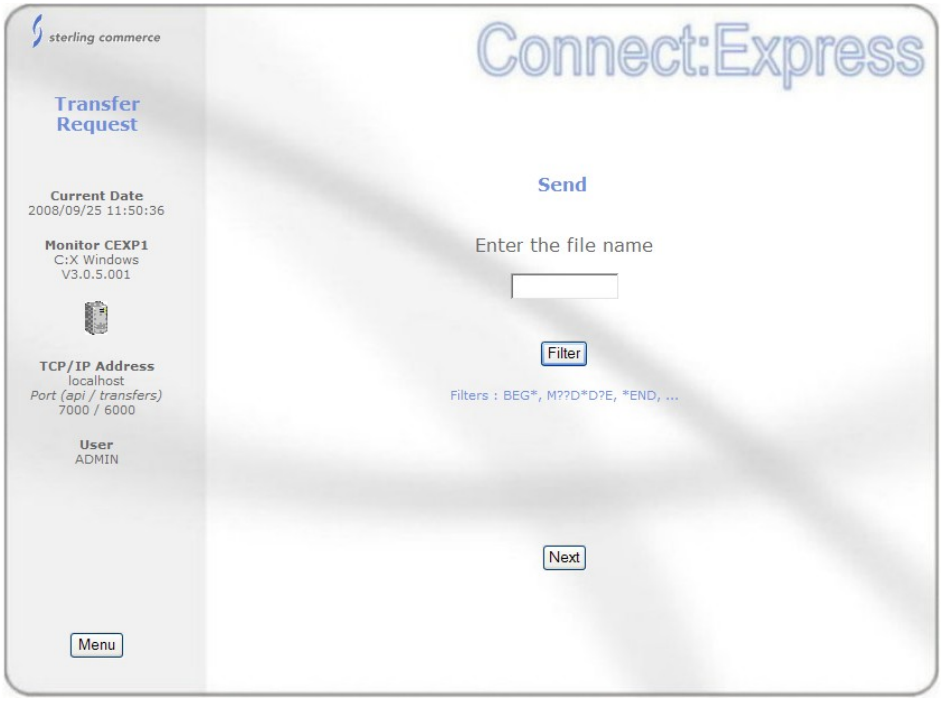
- Filter = define a new filter
- Previous = previous screen
- Return = return to the first screen of the current process
- Cancel = cancel the current process

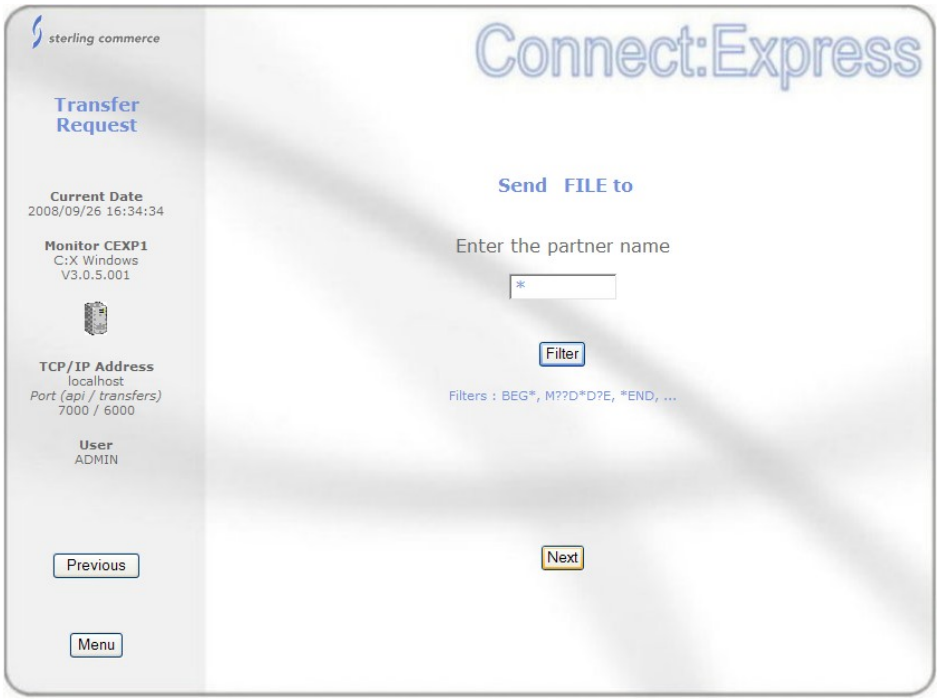


Session and Presentation tables are not supported by this interface : use the Standard operator interface of the product.

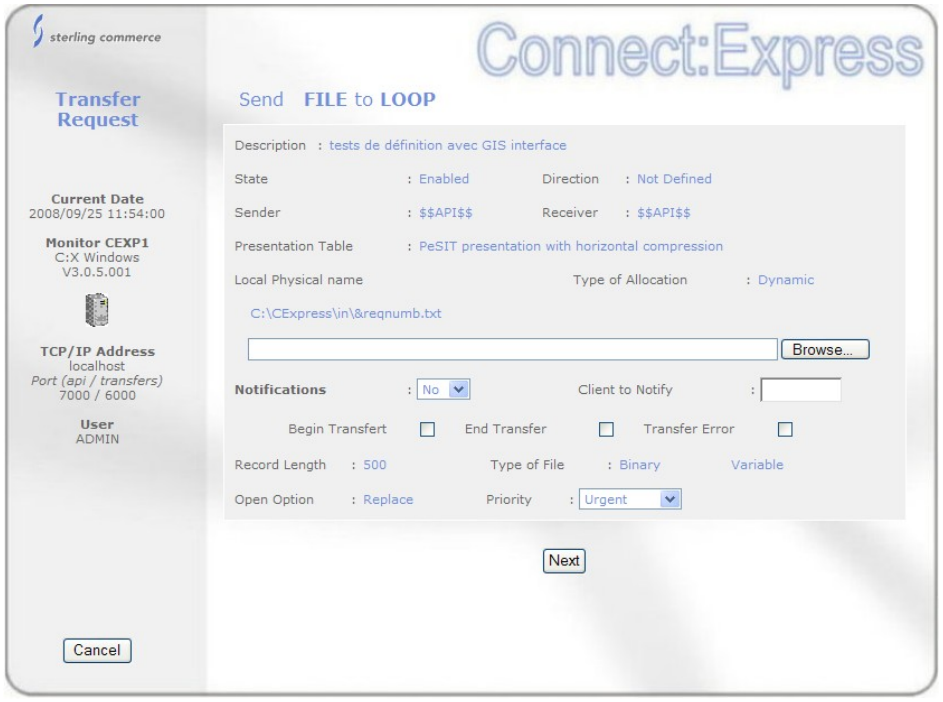
File Transfer Request

You can send or receive a file : you are first asked for the name of the file. If the file definition matches the direction, you can provide the name of the partner. However, if the name of the partner is defined in the definition of the file, the interface skips this step and goes directly to next screen.

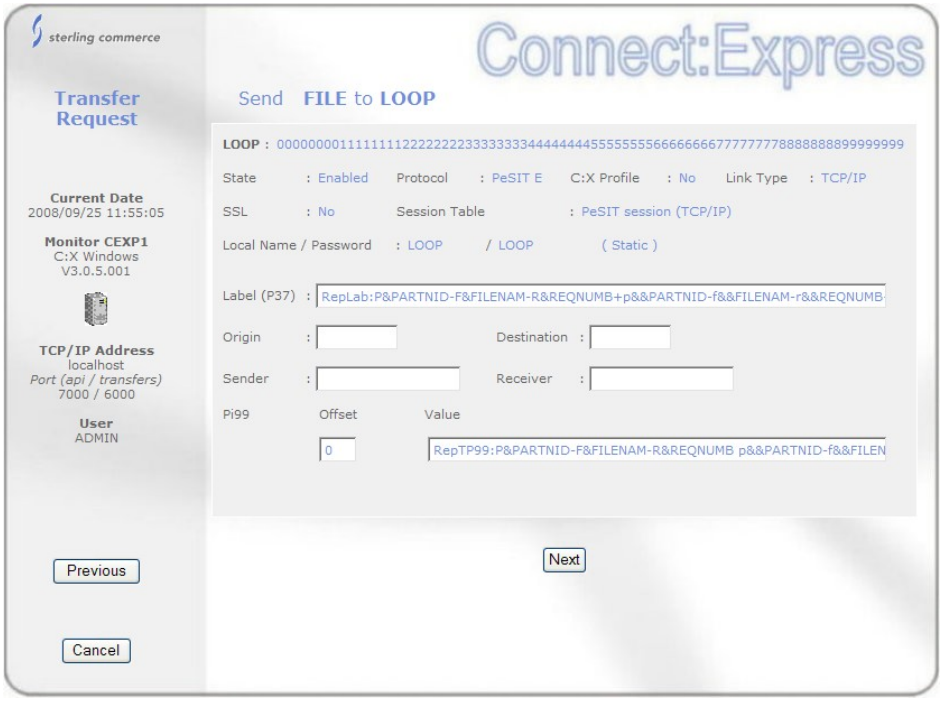




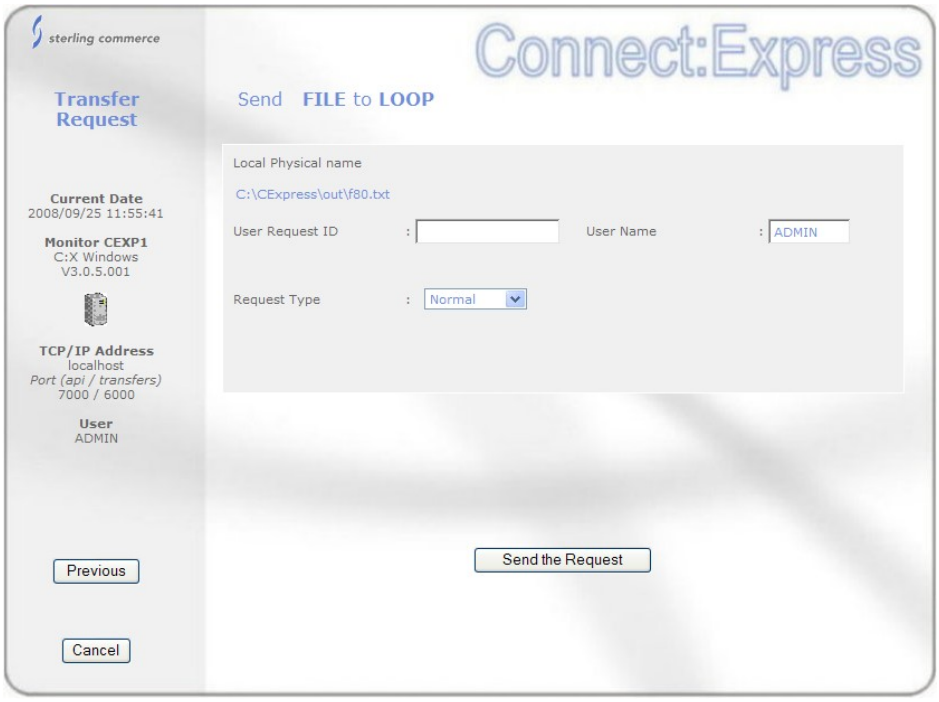
Once you have provided the file and partner names, you can provide more parameters, first from the file definition, then from the partner definition, and finally parameters specific to the request.



File parameters

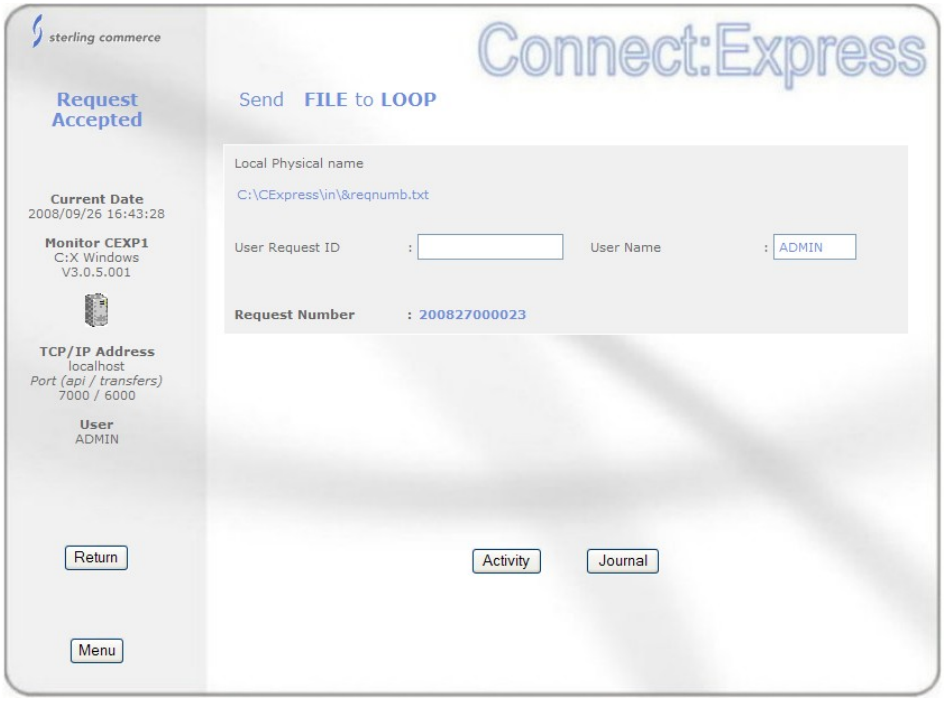


Partner parameters – depending on the protocol



Request parameters

When the request is accepted, you request number is shown and you can access to Activity and Journal functions.



Monitoring Requests

You can monitor requests in two ways : Activity or Journal. Activity consists of requests that are being processed or eligible to processing, Journal consists of requests ended successfully or not.

The list of status considered as active depends on the platform.

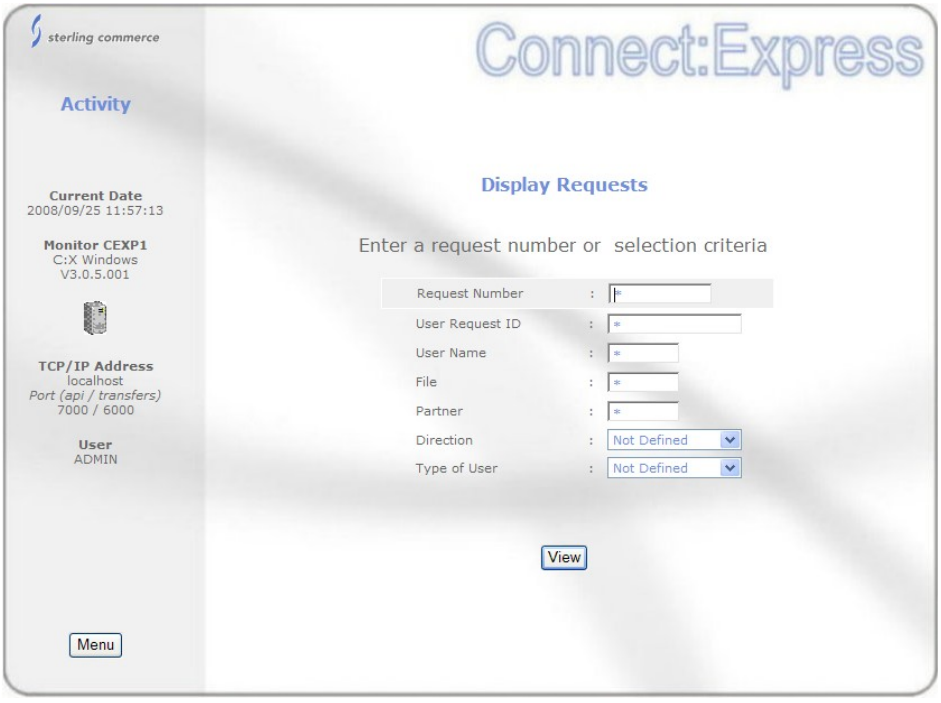
On Windows :

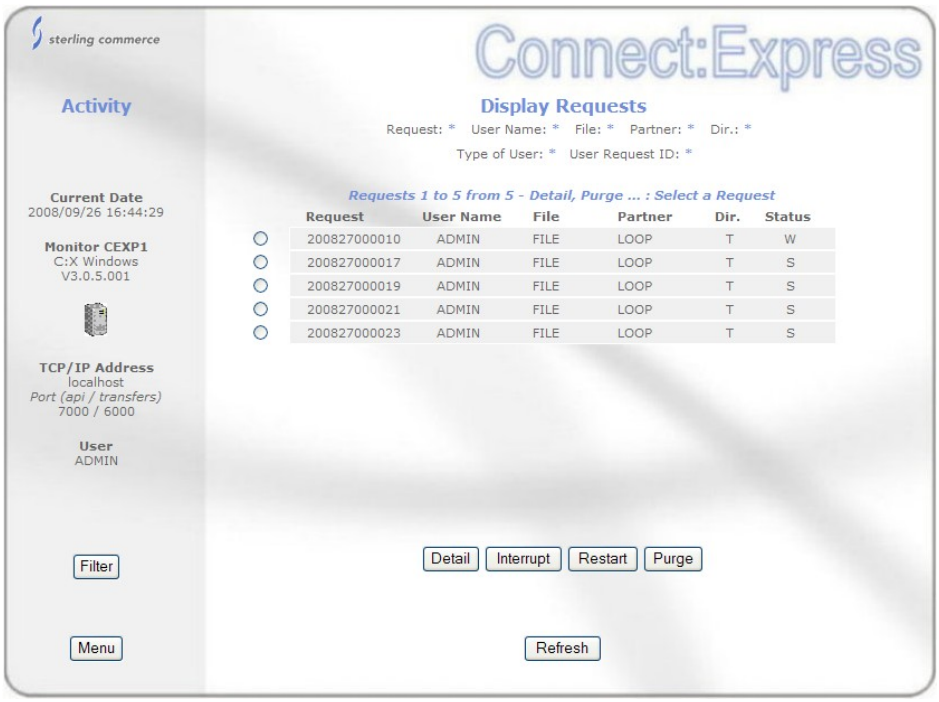
· W : Waiting selection
· T : Selected
· C : Running
· I : Interrupted
· S : Selection error
· R : Restarting

On Unix :

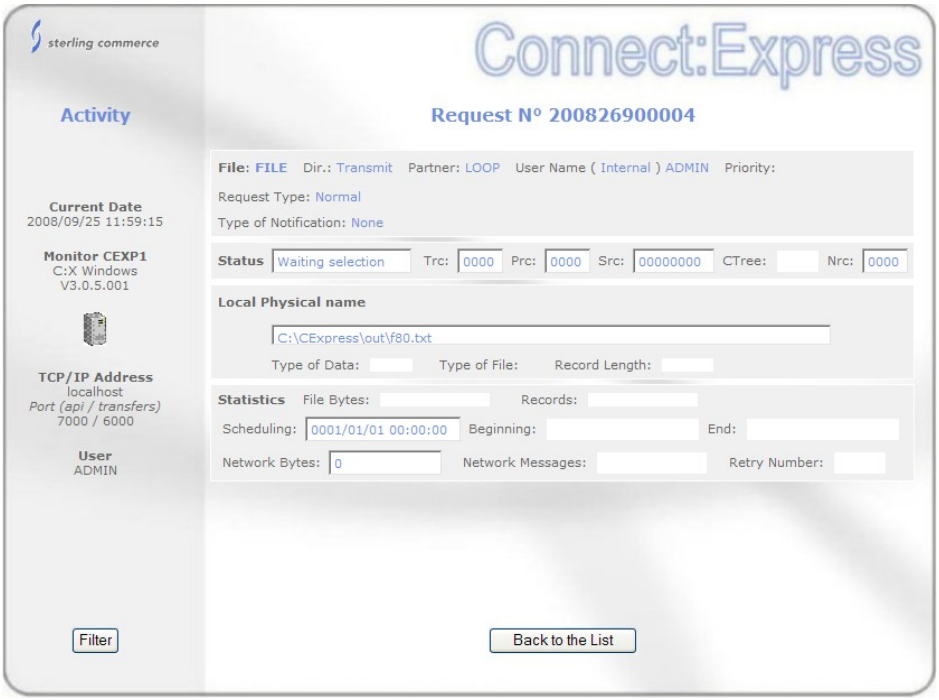
· A : Waiting selection
· C : Running
· O : Interrupted
· D : Differred
· J : Restarting
· K : Waiting restart

Activity function provide a filter as shown in the figure below. You can either provide one request number or one or several criteria.





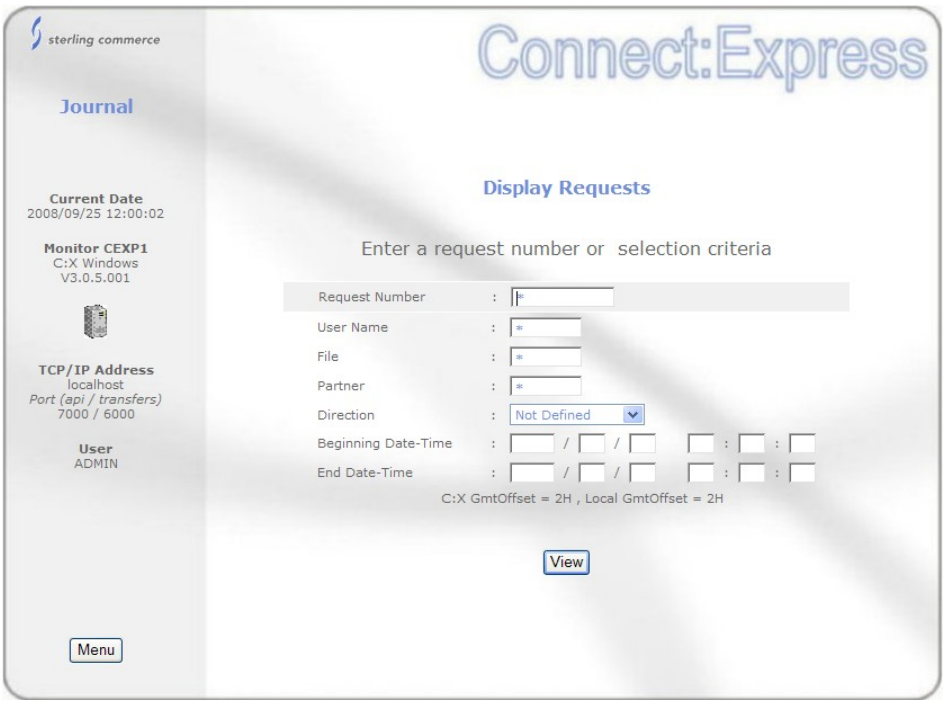
The list screen displays the active filter in the upper part. If you want to display information of a request, select it and click on *Detail*. You can also *interrupt* a started request, *restart* an interrupted request or *purge* a request. You can *refresh* the list and update the status.



Request information is displayed in one screen.

You can display journal records using a filter as shown in the following figure. You can either provide one request number or one or several criteria.

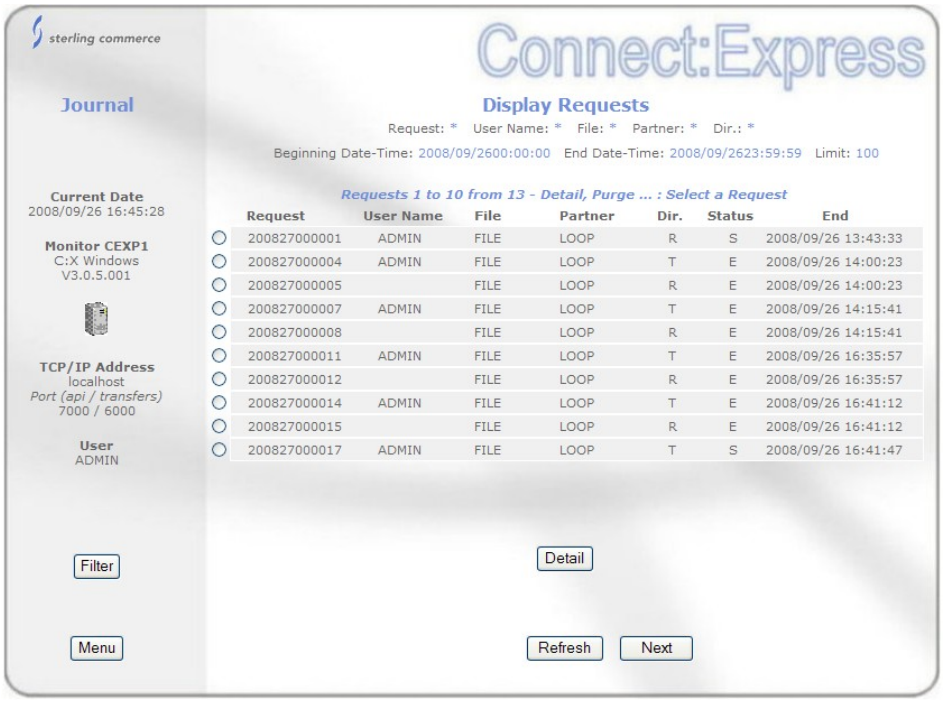
The default time interval, in the format '*AAAAMMJJ HHMMSS*', is the current day, for the interface environment. The screen shows the GMT offsets in the monitor environment (C:X GmtOffset) and the GMT offsets in the interface environment (local GmtOffset) because the time interval must be considered in the monitor environment : if the two offsets are equal, just consider your time.

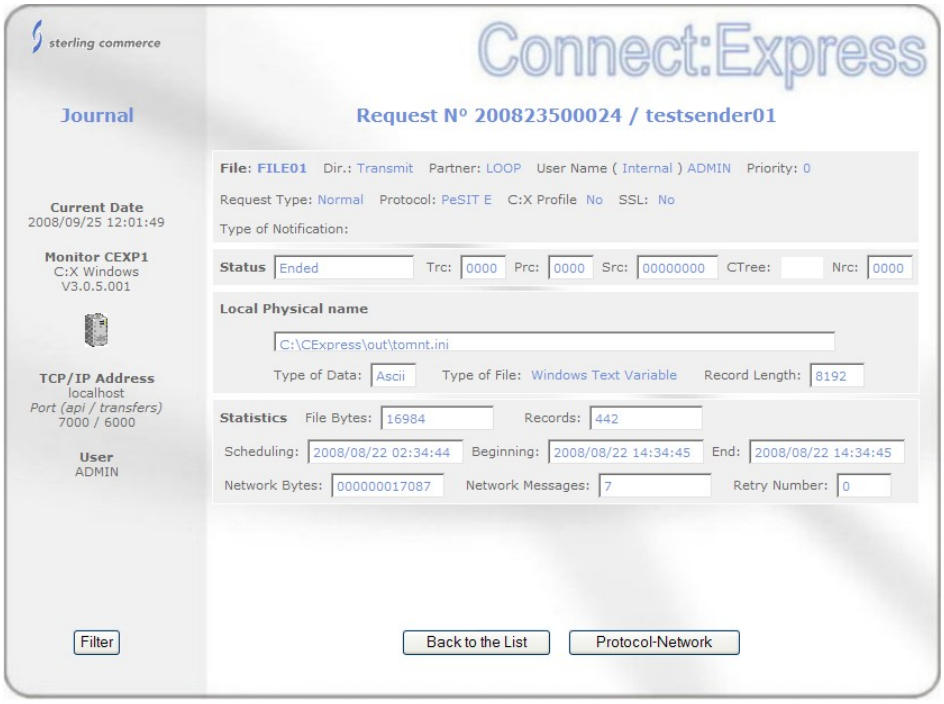


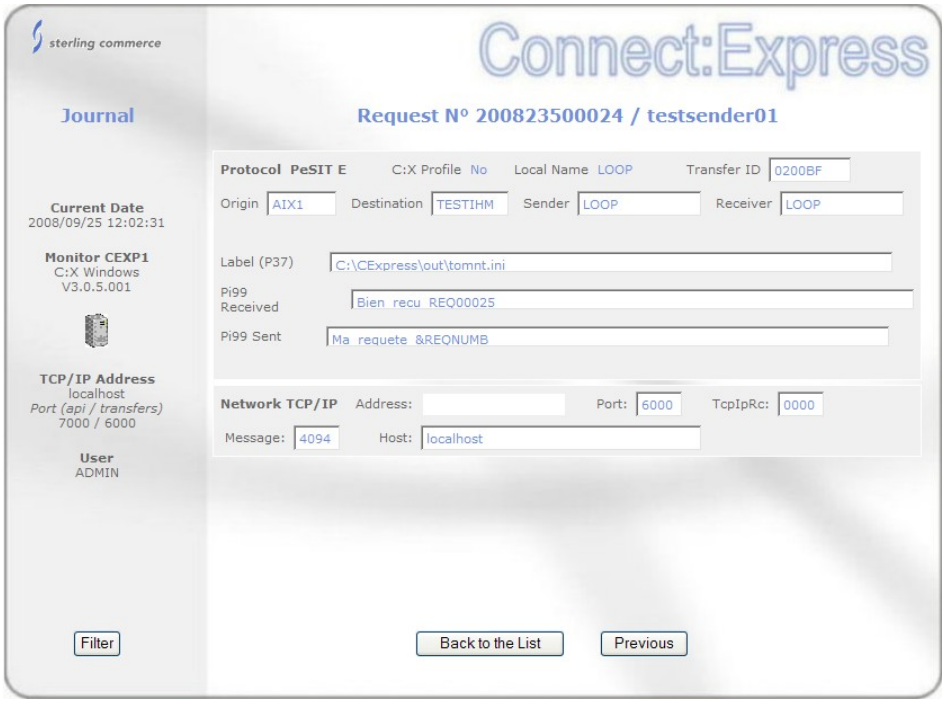
The list screen displays the active filter in the upper part. It shows that the list is limited to 100 requests. If the time interval corresponds to more than 100 requests, you might have to modify the beginning and end date to obtain the 100 requests you want.

If you want to display information of a request, select it and click on *Detail*. You can *Refresh* the list and update the status.

Request information is displayed in two screens: the first screen displays the results of the transfer, the second screen displays information that depend on the protocol


sterling commerce
Connect:Express
Journal
Display Requests
Request: * User Name: * File: * Partner: * Dir.: *
Beginning Date-Time: 2008/09/2600:00:00 End Date-Time: 2008/09/2623:59:59 Limit: 100
Current Date
2008/09/26 16:45:28
Monitor CEXP1
C:X Windows
V3.0.5.001
TCP/IP Address
localhost
Port (api / transfers)
7000 / 6000
User
ADMIN
Requests 1 to 10 from 13 - Detail, Purge ... : Select a Request
Request User Name File Partner Dir. Status End
200827000001 ADMIN FILE LOOP R S 2008/09/26 13:43:33
200827000004 ADMIN FILE LOOP T E 2008/09/26 14:00:23
200827000005 FILE LOOP R E 2008/09/26 14:00:23
200827000007 ADMIN FILE LOOP T E 2008/09/26 14:15:41
200827000008 FILE LOOP R E 2008/09/26 14:15:41
200827000011 ADMIN FILE LOOP T E 2008/09/26 16:35:57
200827000012 FILE LOOP R E 2008/09/26 16:35:57
200827000014 ADMIN FILE LOOP T E 2008/09/26 16:41:12
200827000015 FILE LOOP R E 2008/09/26 16:41:12
200827000017 ADMIN FILE LOOP T S 2008/09/26 16:41:47
Filter
Detail
Menu
Refresh
Next


sterling commerce
Connect:Express
Journal
Request N° 200823500024 / testsender01
Current Date
2008/09/25 12:01:49
Monitor CEXP1
C:X Windows
V3.0.5.001
TCP/IP Address
localhost
Port (api / transfers)
7000 / 6000
User
ADMIN
File: FILE01 Dir.: Transmit Partner: LOOP User Name (Internal) ADMIN Priority: 0
Request Type: Normal Protocol: PeSIT E C:X Profile No SSL: No
Type of Notification:
Status Ended Trc: 0000 Prc: 0000 Src: 00000000 CTree: Nrc: 0000
Local Physical name
C:\CExpress\out\tomnt.ini
Type of Data: Ascii Type of File: Windows Text Variable Record Length: 8192
Statistics File Bytes: 16984 Records: 442
Scheduling: 2008/08/22 02:34:44 Beginning: 2008/08/22 14:34:45 End: 2008/08/22 14:34:45
Network Bytes: 000000017087 Network Messages: 7 Retry Number: 0
Filter
Back to the List
Protocol-Network

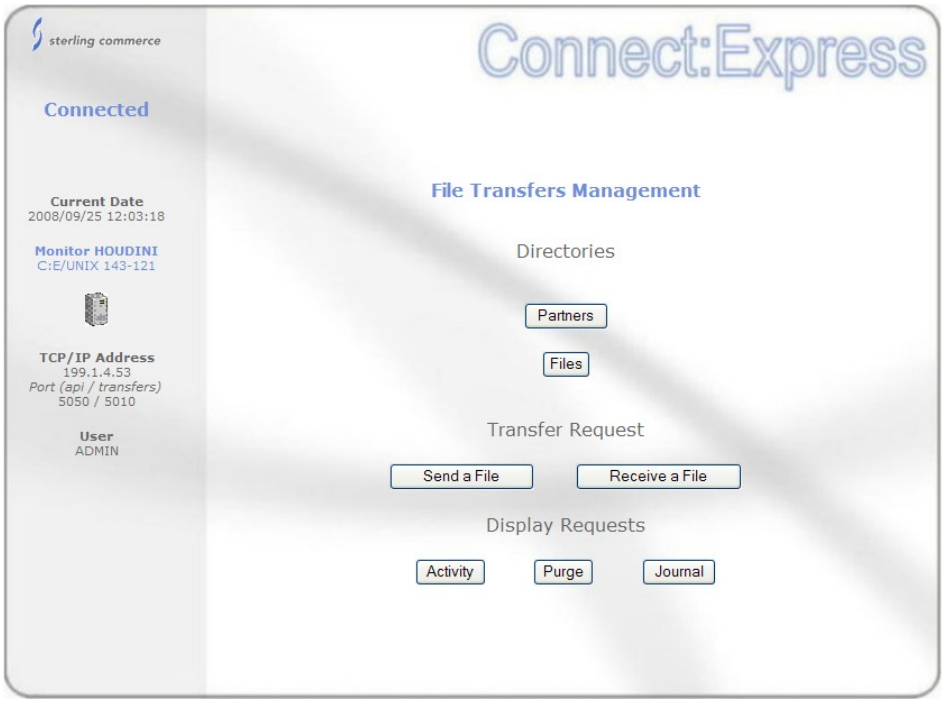


Purging Requests

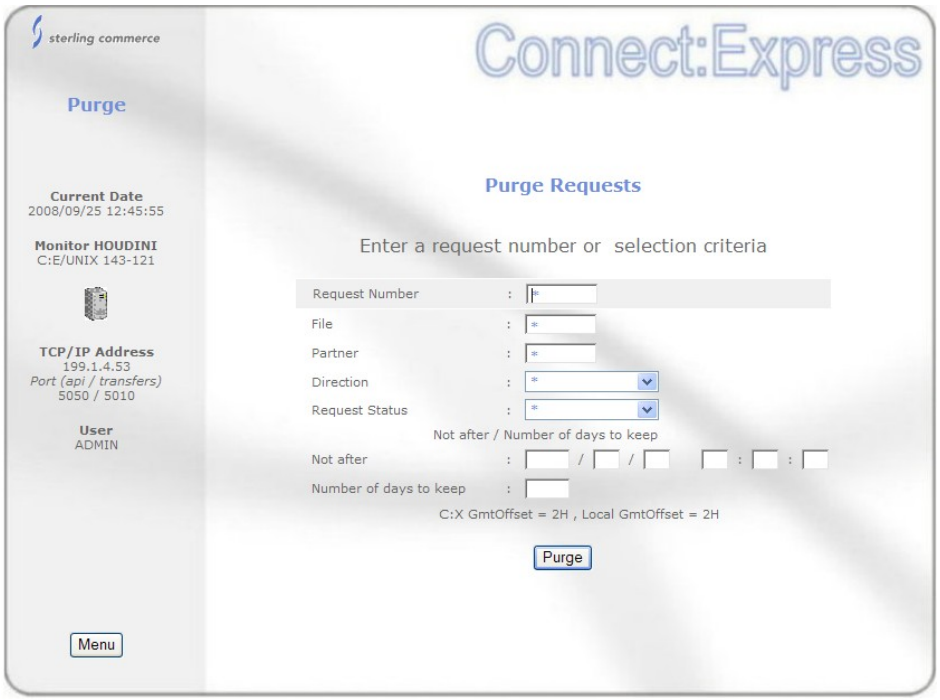
The purge request functionality depends on the platform were the monitor executes.

For Connect:Express Windows and Unix, the purge function is available in the Activity screen, along with interrupt and restart functions.

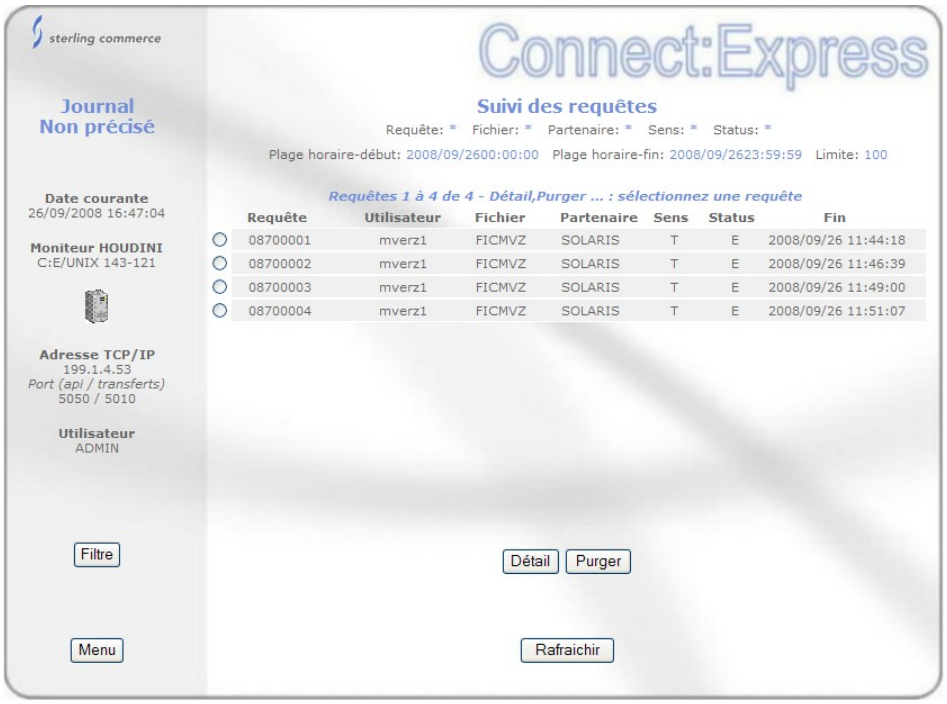
For Connect:Express Unix, the purge function is also available in the Journal screen : it works the same way as in the Activity screen. On an other hand, you are provided a purge function on the Menu screen: this function enables you to purge one request or several requests from criteria. Among the criteria, you can provide either the date after which requests must be kept or the number of days to keep. These two parameters exclude each other.



Purge function from the menu for Unix



Criteria for purge on Unix



Purge function from the journal screen for Unix

Integration with SI

The description above is available whatever the implementation is, with or without integrating it in an application. You can integrate the interface in a SI application passing parameters to a business process through the URI : the BP is launched during the login phase.

The default configuration of the interface works without SI control : you can use parameters to implement controls.

Parameters of the URI

The parameters of the URI enable you to determine the way the interface will initialize and to provide all or part of the parameters to access the monitor.

There are three ways to initialize :

- Interfacing directly with Connect:Express : this is the default as described above. You can provide parameters to access the monitor. No login phase is activated, no business process is launched.
- Login to Connect:Express : only available with Connect:Express Windows, this mode of initialization requires the user to login to the monitor. A Connect:Express login is activated. No business process is activated.
- Login to SI : in this mode, the user is required to login to SI. The standard SI login screen is activated and the Connect:Express configuration business process is launched.

The syntax to use is discussed below :

```
CXSession? [cxconfigbp=xxxxx/NO] [&cxlogin=YES/NO]
           [&cxuser=xxxxx/ADMIN ][&cxpassword=xxxxx/ADMIN ]
           [&cxipaddr= xxx.xxx.xxx.xxx/localhost] [&cxipport=ppppp/7000]
          [&cxostype=WINDOWS/UNIX] [&lang=FR/EN]
```

cxconfigbp	NO (default): direct call to Connect:Express, no SI session, input Connect:Express parameters provided. xxxxx: name of a business process to launch. An example is provided: *CXConfig*. The BP receives the Connect:Express parameters input, completes or modifies them, returns to the calling jsp page. This parameter is overrided by cxlogin=YES.
cxlogin	YES: direct call to Connect:Express, no SI session, after login to Connect:Express Windows (user/password) NO (default): Connect:Express identification is provided in the parameters or equals to default (C:X Windows) cxlogin=YES iverrides cxconfig parameter.
Cxuser Cxp assword	Identification parameters to Connect:Express Windows. Default is *ADMIN* for both parameters.
Cxipaddr Cxipport Cxostype	Parameters to access Connect:Express. Default address is *localhost*, default port is *7000*, default OS is the OS where the interface executes (Windows more generally)
lang	FR (default): screens are in French. US, EN … screens are in English

Example 1 :
http://si:10000/CXSession/?cxconfigbp=CXConfig&cxipaddr=199.1.4.53&cxipport=5050&cxostype=WINDOWS

Calling the URI CXSession this way results in the following:

- The user logs in to SI
- The Business process CXConfig is launched: it receives input the characteristics of Connect:Express and the name of the SI user. The Business process role is to define the identification parameters to Connect:Express WINDOWS for this SI user.
- If this step is executed successfully, communications with Connect:Express can start. The menu is displayed and all consecutive operations are executed without launching any Business process.

Example 2 :
http://si:10000/CXSession/?cxlogin=YES

Calling the URI CXSession this way results in the following:

- The user logs in to Connect:Express Windows
- The interface initializes the process by acquiring the monitor configuration. Because no Connect:Express parameters are provided with the URI, le monitor is called at address = localhost, port = 7000, OS = the OS

where the interface is executing (probably WINDOWS).

- If the login is accepted by the monitor, the menu is displayed and all consecutive operations are executed without launching any Business process.

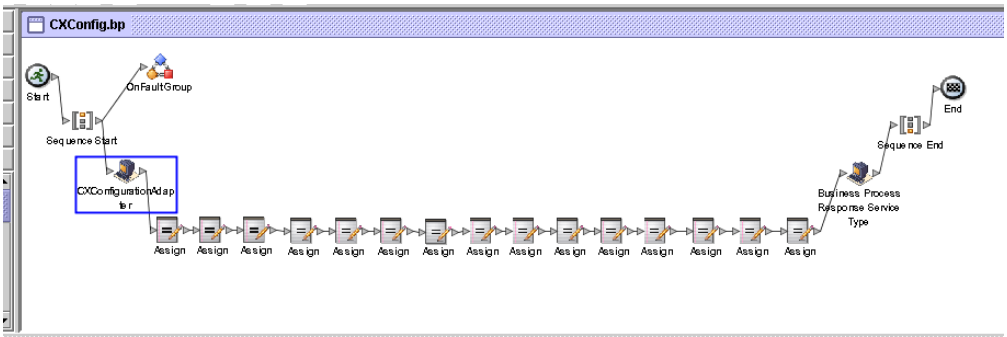
Configuration Business Process

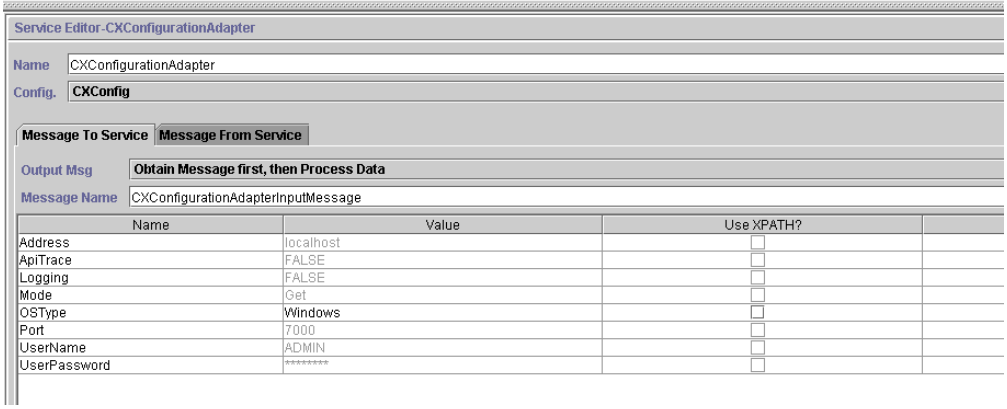
The Business process role is to provide all or part of the parameters to access the monitor. The example of BP provided defines all parameters : this means that only one monitor can be accessed through this Business Process, and the Connect:Express user is unique.

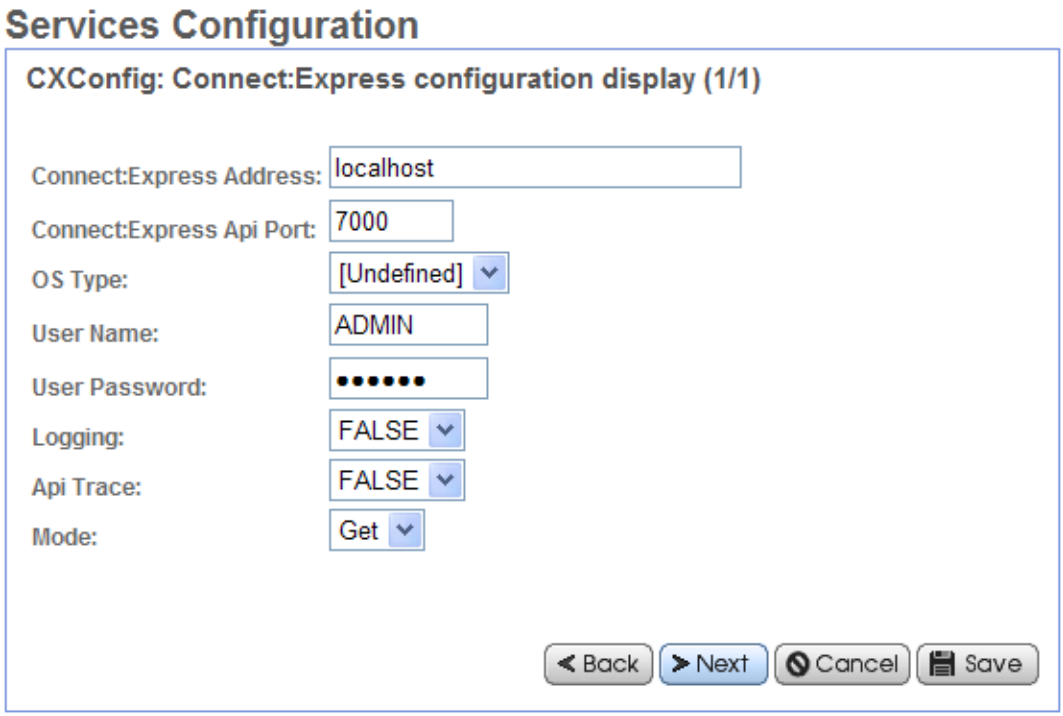
The monitor is defined by *CXConfig* configuration of CXConfigurationAdapter service : CXConfig 'defines' the monitor *localhost:7000*, user *ADMIN*, password *ADMIN*.

The user logged in to SI as *admin.*

The business process inquires the monitor configuration and returns the information to the calling jsp page. This information is shown on the left part of the screen.







Communication structures with the Business process are shown below.

The business process receives input the parameters provided by a jsp page.

Parm = `CXUser=CXUser&CXPassword=CXPassword&CXIpaddr=CXIpaddr&CXIpport=CXIpport&CXOsType=CXOsType`

```
runBP user="username" nvp=parm pridoc="" bpname="CXConfig">
```

The input structure is :

```
<ProcessData>
  <PrimaryDocument SCIObjectID="Gis:fa7e74:11c795e7d84:-6aa0"/>
  <CXIpport>7000</CXIpport>
  <system-account-user-id>admin</system-account-user-id>
  <CXOsType>WINDOWS</CXOsType>
  <CXIpaddr>localhost</CXIpaddr>
  <CXUser>ADMIN</CXUser>
  <CXPassword>ADMIN</CXPassword>
  <username>admin</username>
</ProcessData>
```

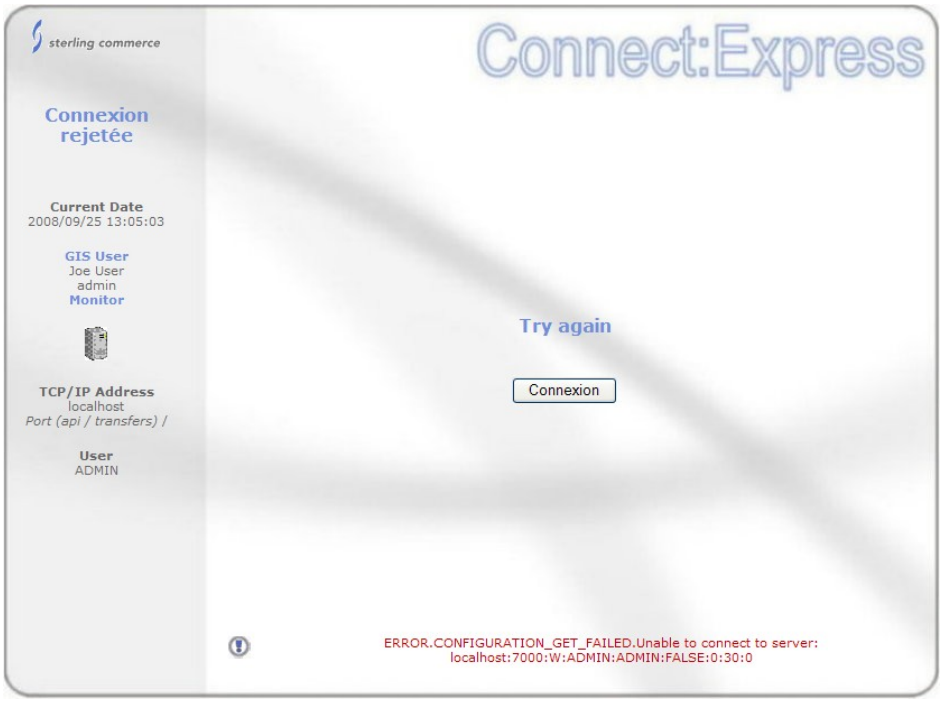
The expected output structure is :

```
<CXResult>
  <CXServerId>localhost:7000:W:ADMIN:****:FALSE:0:30:0</CXServerId>
  <Dpcsid>CEXP1</Dpcsid>
  <CXUser>ADMIN</CXUser>
  <TcpipListenAddress>localhost</TcpipListenAddress>
  <CXOSType>WINDOWS</CXOSType>
  <TcpipListenPort>6000</TcpipListenPort>
  <ApiPort>7000</ApiPort>
  <ProductInfo>C:X Windows V3.0.5.001</ProductInfo>
  <SystemInfo>Windows XP</SystemInfo>
  <BPResponse>Connected</BPResponse>
</CXResult>
```

<BPResponse> field stands for the return code : the character string "Connected" is for success. Any other string means that the connection failed.

```
<CXResult>
  <CXServerId>localhost:7000:W:ADMIN:****:FALSE:0:30:0</CXServerId>
  <CXUser>ADMIN</CXUser>
  <TcpipListenAddress>localhost</TcpipListenAddress>
  <CXResponse>ERROR.CONFIGURATION_GET_FAILED.Unable to connect to
     server: localhost:7000:W:ADMIN:ADMIN:FALSE:0:30:0</CXResponse>
  <BPResponse>Connection rejected</BPResponse>
</CXResult>
```

CXResponse fields contains the error message from Connect:Express interface : The browser interface shows this message in the bottom.



Using a Business Process enables you to limit the number of parameters you provide to the URI. For example you can define one BP for one monitor and pass no parameter in the URI. You can define one BP for one user and pass the monitor parameters through the URI.