



# **IPDR/File Transfer Protocol**

**Version 3.5.0.1**

**November, 2004**

**© 1999-2004 IPDR.org, Inc.**

## Table of Contents

1	INTRODUCTION .....	3
2	ARCHITECTURE .....	4
2.1	Groups.....	4
2.2	Group Sequence Numbers .....	4
2.3	Subscriptions.....	4
2.4	Document Ids.....	5
2.5	Control files and Control file Rolling.....	5
2.6	The Capability File .....	6
2.7	Range Files .....	6
2.8	Aging of IPDR Documents and Control files.....	6
3	MAPPINGS FROM TRANSFER PROTOCOL TO FILE.....	7
	SHARING IMPLEMENTATION	
3.1	IPDR Document Files.....	7
3.2	Subscription-group Control-files .....	7
3.3	Control File Rolling and Aging .....	8
3.4	Range file.....	9
3.5	Capability File Annotated Description .....	9
3.5.1	<groupId> .....	10
3.5.2	<controlFileDirectory> .....	10
3.5.3	<controlFilePrefix>.....	10
3.5.4	<controlFileNamePolicy>.....	10
3.5.5	<controlFileSuffix> .....	11
3.5.6	Sample Capability File.....	11
4	SECURITY CONSIDERATIONS .....	12

## 1 Introduction

The abstract IPDR Transfer Protocol provides three models of delivery:

- IT Push, where the IPDR Transmitter is responsible for delivering IPDR documents to a known Business Support System (BSS).
- BSS Pull, where a BSS requests specific documents from an IT.
- Demand Poll, where the IT system notifies the BSS of the availability of IPDR documents and the BSS is responsible for pulling them.

The Transfer Protocol is defined in a transport neutral manner. The working groups of IPDR.org will define specific mappings.

This document describes a file based transfer protocol. This file based mapping is a minimal implementation of the Pull model. For the file transfer protocol mapping, a subset of capabilities are supported and communication is done through simple URL's which can indicate a file is accessible via the local file system, FTP or HTTP(S).

Real time streaming protocols and a web services mechanism for exchange are defined in other IPDR 3.5 documents.

The following sections define the Protocol:

- "Groups, Sequence numbers, Document ids and Subscriptions" describes how the IPDR Transmitter organizes IPDRDocs for retrieval and delivery to BSSs. The protocol primitives are dependent on these concepts.
- "File Transfer Protocol Mapping," describes the file based transfer protocol used in several IPDR interoperability demonstrations. Note that this is now considered an instance of the defined protocol (albeit a trivial one).
- "Security Considerations", describes the security mechanisms available with the mapping.

## 2 Architecture

An IPDR Transmitter organizes IPDRDocs into groups. A sequence number and a unique document id identify the documents in a group. Consuming systems access IPDRDocs by explicitly pulling specific documents.

### 2.1 Groups

An IPDR Transmitter will receive either IPDR Documents from an IPDR Store or individual IPDRs from one or more IPDR Recorders. If individual records are received, it is the responsibility of the IPDR Transmitter to assign sets of records to an IPDRDoc and to place them in a logical IPDR Store.

Each IPDR Document the IPDR Transmitter must process is assigned to zero or more groups. The business rules, which assign IPDR Documents to groups, are a local matter for the IPDR Transmitter. Similarly the set of groups maintained by the IPDR Transmitter is a local matter.

Groups are named with a string identifier. The assignment of group names are a local matter, the only restriction is that an IPDR Transmitter with multiple groups must have a unique name for each group. Two different IPDR Transmitters may use the same group name, but no relationship between these groups is implied.

### 2.2 Group Sequence Numbers

When the IPDR Transmitter assigns an IPDR Document to a group the document is given the next available sequence number for that group. Sequence numbers within a group are never repeated. To guarantee this property 64-bit integers may be needed to prevent roll over.

Sequence numbers are positive integers that increase by one for each new document. When a new group is created, the first document assigned to that group should be given the sequence number of 1.

As IPDR Documents are aged off based on policy, the corresponding sequence number within that group is no longer considered available.

### 2.3 Subscriptions

Because of the passive nature of the IPDR/File Transfer Protocol, the IPDR Transmitter does not maintain subscriptions. Access control to the file system or other means should be used to limit which systems may access the IPDR Documents. But aside from basic authentication credentials, the IPDR Transmitter holds no state for the different consumers, in the case of IPDR/FTP.

The consumer may maintain logical subscriptions.

As an IPDR Producer creates IPDR documents (abbreviated as IPDRDocs) it places them into well formed and valid IPDR/XML or IPDR/XDR encoded files. The IPDR Producer sorts each

IPDRDoc into one or more groups. IPDRDocs will often be grouped by service type, by customer name, or by a combination of the two, though the IPDR Producer is also allowed to use any other useful policy to sort IPDRDocs into Groups. An IPDR/Capability File identifies the available groups. The groups provide an easy way for an IPDR Consumer to access only those subsets of IPDRDocs that are of interest to it.

## 2.4 Document Ids

The root <IPDRDoc> element in all IPDR Documents must have a globally unique identifier associated with it. These identifiers use Universal Unique Identifiers (UUIDs). UUIDs are useful in that they provide a means to guarantee the generation of unique identifiers across a distributed system without requiring central administration of the identifiers, which can be costly.

This identifier is assigned when the IPDR Document is created. When the IPDR Transmitter assigns a document to a group, the document's Id is not modified. Rather the document is assigned a sequence number within that group. If the same document is assigned to multiple groups, the sequence number assigned for each group is unrelated. However, the document maintains its single unique document identifier, so that instances of the same document in multiple groups can be identified.

The definition of UUIDs resides in the OSF specification for Distributed Computing Environment (DCE). OSF CAE Specification, Document C706, 1997, Appendix A, located at:

<http://www.opengroup.org/onlinepubs/009629399/>

UUIDs are equivalent to Microsoft's Globally Unique Identifiers (GUIDs).

An open source C implementation of UUID generation is available in the appendix of the IETF draft, draft-leach-uuids-guids-01.txt. This draft has expired, but an archived copy is available at:

<http://www.ipdr.org/public/draft-leach-uuids-guids-01.txt>

Note: the IETF draft was allowed to expire because the group considered the OSF work a referenceable standard and did not choose to duplicate it.

## 2.5 Control files and Control file Rolling

As the IPDR Producer places an IPDRDoc into a subscription group, it adds the name (and possibly the location) of the IPDR document file to the last line of the current control file for that subscription group. The names of the control files follow a predictable naming convention, which is <prefix><sequenceNumber><suffix>. The IPDR Producer may periodically create a new control file and name it by incrementing the current sequence number by one. This process of periodically switching to a new control file is referred to as "rolling". When the Producer does this, it places a special end-of-file string at the end of the previous control file, so that any consumers who are reading from the subscription group will know to stop reading the old control file and to open the new control file. A common rolling policy is one that uses date-time stamps,

such that the IPDR Producer would change control files every hour or every day. However, the IPDR Producer is free to implement any “rolling” policy.

## 2.6 The Capability File

The IPDR Producer maintains a capability file, which has a list of all the subscription groups. The capability file contains all the information an IPDR Consumer needs to “subscribe” to one or more subscription groups. In this file sharing implementation, subscribing simply involves the IPDR Consumer copying or ftping the IPDRDocs from a subscription-group. In this file-sharing scheme, the IPDR Producer has no knowledge of which IPDR Consumers are reading its IPDRDocs, nor does it know which IPDRDocs have been accessed.

## 2.7 Range Files

All of the control files for a specific subscription group are contained in one directory and the path to this directory is in the capability file. The control file directory also contains a range file for that subscription group. It may also contain the IPDR document files to which the control files refer but this is not required. The range file is used IPDR Consumers subscribed to the subscription-group to determine which control files currently exist for that subscription group, and most importantly which control file is currently having the names of new IPDR document files appended to it. The range file contains two sequence numbers separated by a dash. The first is the sequence number of the oldest control file. The second is the fixed-length sequence number of the current control file. At times, if the size of the sequence number isn’t large enough to handle the number of control files, the sequence number may reach its maximum value and then roll back to all zeroes. So it is possible that the left-hand sequence number in the range file may be larger than the right-hand sequence number. An IPDR Consumer that has just subscribed to a subscription group would read the range-file for that group to get the sequence number of the oldest control-file. Using that sequence number, the IPDR Consumer could determine the name of the oldest control-file and start reading the IPDRDocs to which that file refers. By consecutively incrementing the sequence number, it could then read through all control-files in order until it reached the end of the most recent one.

## 2.8 Aging of IPDR Documents and Control files

Multiple control files are used to make it easy for the IPDR Producer to periodically remove old IPDRDoc, a process known as “aging”. Using the range file the IPDR Producer can easily identify the oldest control files and remove those control files as well as the IPDRDocs to which those control files refer. When removing old IPDR documents and control files, the range file would also be updated to reflect the recent deletions. The IPDR Producer is free to implement any aging policy. One common aging policy would involve checking the control files on a daily basis, and deleting all control files that are more than a week old. However, in high volume environments the check might be done on an hourly basis and the IPDR Producer might remove all files that are more than a day old. The file sharing implementation of the transfer protocol provides an infrastructure that supports a variety of aging policies, so the choice is up to the IPDR Producer.

### 3 Mappings from Transfer Protocol to File Sharing Implementation

There are no explicit primitives for CapabilityReq, ListGroupsReq, and ListGroupsRsp; however the functions performed by these primitives can still be accomplished within the file sharing implementation. Instead of issuing a ListGroupsReq and waiting for the ListGroupsRsp, the IPDR Consumer can get a list of subscription groups by reading the capability file and reading the <groupInfoList> and <groupInfoItem> elements. Rather than exchanging ListDocsReq and ListDocsRsp messages to get a list of all the documents in a subscription group, the IPDR Consumer reads the range file and control files for that subscription group. The IPDR Consumer subscribes to a subscription group simply by monitoring available files without the explicit knowledge of the Producer, so there are no analogues in the file-sharing mapping to the SubscribeReq, SubscribeRsp, UnsubscribeReq, UnsubscribeRsp, PushReq, or PushRsp primitives. The IPDR Consumer accomplishes a PullReq/PullRsp simply by copying or ftping an IPDRDoc referred to in a control file for a particular subscription group.

#### 3.1 IPDR Document Files

Each IPDR document must be placed in an ASCII file. Each file must contain exactly one IPDRDoc. These IPDR document files are not required to follow any specific naming convention, however, each IPDR document file name must be unique. The IPDR Producer is responsible for ensuring that no two IPDR documents are created with identical names. It may be useful to incorporate such information as the IPDR Producer name, a date/time stamp, and/or the IPDRDoc UUID into the file name.

#### 3.2 Subscription-group Control-files

Each subscription group has one or more control files. Each control file is an ASCII file that lists a subset of file names corresponding to the IPDR documents included in that subscription group. The first line of each control file must always be "VERSION 3" followed by a linefeed character.

Each subsequent line of the control file contains either the URL or the file name of exactly one IPDR document file. If just the IPDR document file name is used, it is assumed to reside in the same directory as the control file that references it. Each line of text in the control file should be terminated with exactly one linefeed.

The IPDR Producer must append a "VERSION 3" to the end of the control file to denote that no more IPDR document filenames will be added to that control file.

The maximum number of lines in a control file is unconstrained. The minimum number of lines in a control file is 1 while the control file is still "active" (i.e., the IPDR Producer is still adding IPDR document filenames to that control file), and 2 once the control file is no longer having new IPDR document filenames added to it. A minimal control file, that is no longer active would look like this (where <EOF> denotes the end-of-file):

```
VERSION 3  
VERSION 3
```

<EOF>

The name of the control file is required to follow the naming convention specified in the IPDR Producers capability file. The control file name has unchanging prefix and suffix, which are defined in the capability file using the <controlFilePrefix> and <controlFileSuffix> elements. In between these there must be a sequence number. The sequence number is a decimal integer with the number of digits specified by the <controlFileNamePolicy> in the capability file. The sequence number part of the control file name must always have the number of digits specified in the capability file and must use leading 0's to effect this.

The presence of an IPDR document filename or URL in the control file implies that the named IPDR Document file must exist. The IPDR Producer is responsible for ensuring that files cited in the control files are actually available at the referenced location.

The IPDR Producer is responsible for ensuring that control file references point only to valid, fully formed, completed (no longer changing) IPDR documents. This implies that the IPDR Producer should finish writing to an IPDR document file before appending that files name to the control file, and should delete control files before deleting IPDR documents.

When reading the control file, the IPDR Consumer is required to open it in read-only mode. When adding new IPDR documents to a subscription-group, the IPDR Producer should open the control file in append mode.

### 3.3 Control File Rolling and Aging

In production environments, a single control file is insufficient because over time it will grow very large. In addition this would imply that the referenced IPDR document files are kept indefinitely (since they are referenced by the control file).

An optional rolling policy can be employed to indicate that a new control file will be created periodically. The use of the rolling option requires that control file names contain a sequence number embedded between the prefix and the suffix. The number of digits in that sequence number shall be held constant and the numeric value shall rise incrementally (recycling to 0 upon hitting the highest value).

If the <controlFilePrefix> was "voip1\_IT1\_", the <controlFileNamePolicy> was "NNNNNN", and the <controlFileSuffix> was ".log", then the control file names would be generated in the following order:

```
voip1_IT1_000000.log
voip1_IT1_000001.log
voip1_IT1_000002.log
...
voip1_IT1_999997.log
voip1_IT1_999998.log
voip1_IT1_999999.log
voip1_IT1_000000.log
```

voip1\_IT1\_000001.log

...

Note how the sequence number wraps around from 999999 to 000000. This is the only time that the sequence numbers won't be monotonically increasing. It is the responsibility of the IPDR Consumers to sequence number rollover in the particular case where the range file has a left-hand sequence number that is larger than the right-hand sequence number. IPDR Producers are strongly encouraged to choose a sufficient number of digits for the sequence number to prevent rollover from occurring within the time interval in which this file sharing transfer protocol will be used in a production environment.

If the rolling policy is used, the last line of a completed control file shall always be "VERSION 3". This implies that no additional appends will be made to this control file and a higher numbered control file is available; the IT is responsible for ensuring this to be true. Only one control file per group may be open for appending by the IT.

The IPDR Producer may delete old control files and corresponding IPDR document files based on locally defined aging policies; however the list of existing control files must always remain contiguous. For instance, it would be incorrect for an IPDR Producers to delete control file voip-0004.log if both voip-0003.log and voip-0005.log still exist, since that would introduce a discontinuity in sequence numbers.

### 3.4 Range file

Each subscription group must have a range file. The range file is used by both the IPDR Producer and any IPDR Consumers subscribed to the subscription group to determine which control files currently exist for that subscription group, and most importantly which control file is currently having the names of new IPDR document files appended to it. The range file contains two sequence numbers separated by a dash. The first is the sequence number of the oldest control file. The second is the sequence number of the current control file. The name of the range file must be the name of the subscription group, found in the <groupId> element, concatenated with the string "-range-file". The range file for a particular subscription group must reside in the same directory as the control files for that subscription group. This directory is specified by the <controlFileDirectory> element for that subscription group in the capability file.

### 3.5 Capability File Annotated Description

The capability set documented here corresponds to the following entry in the capability file:

```
<supportedProtocolItem version=3.1 protocolMapping="File" encoding="XML">
  <primitiveList>
    <primitiveItem>Pull</primitiveItem>
  </primitiveList>
  <extension>
    <groupInfoList>
      <groupInfoItem>
```

```
</groupInfoItem>
</groupInfoList>
</extension>
</supportedProtocolItem>
```

Each subscription group must have a `groupInfoItem` element containing the following sub-elements:

```
<groupId></groupId>
<controlFileDirectory></controlFileDirectory>
<controlFilePrefix></controlFilePrefix>
<controlFileNamePolicy></controlFileNamePolicy>
<controlFileSuffix></controlFileSuffix>
```

All of these elements are constrained to contain a string containing only the following characters: ‘0-9’, ‘a-z’, ‘A-Z’, ‘.’, ‘-’, ‘:’, ‘/’, and ‘\_’.

### 3.5.1 <groupId>

The `groupId` element must contain the name of the subscription group described by this `groupInfoItem` element. It should be a non-zero length string of allowable characters. Some examples might include: “voip”, “email-1”, and “best-hosting-co”.

### 3.5.2 <controlFileDirectory>

The `controlFileDirectory` contains the URL that points to the directory containing the control files and the range file for the subscription group. A URL is required here to avoid the complications that can arise when local O/S file naming conventions are used. For instance, Microsoft Windows and Unix/Linux operating systems use different characters to denote path’s (Windows uses ‘\’, while Unix uses ‘/’). The URL can either be a “file:” or an “http:”. Some examples might include: “<file:///C:/IPDR/voip1>”, “<http://www.best.com/voip1>”.

### 3.5.3 <controlFilePrefix>

The `controlFilePrefix` element contains a string of allowable characters that denotes the first part of the name of all of the control files in the subscription groups. Some possible examples include: “voip-”, “internet-access\_”, and “best.com.”

### 3.5.4 <controlFileNamePolicy>

The `controlFileNamePolicy` element must contain one or more ‘N’ characters, where the number of ‘N’ characters denotes the number of digits in the sequence number embedded in the name of all control files for this subscription group. Some examples include: “NNNNNN”, and “NNNNNNNNNN”.

### 3.5.5 <controlFileSuffix>

The <controlFileSuffix> element contains a string of allowable characters that denotes the last part of the name of all of the control files in the subscription groups. Some possible examples include: “.log”, “-control” and “”.

### 3.5.6 Sample Capability File

```
<CapabilityRsp>
  <supportedProtocolList>
    <supportedProtocolItem version=3.1 protocolMapping="File" encoding="XML">
      <primitiveList>
        <primitiveItem>Pull</primitiveItem>
      </primitiveList>
      <extension>
        <groupInfoList>
          <groupInfoItem>
            <groupId>sm</groupId>
            <controlFileDirectory> http://www.best.com/sm</controlFileDirectory>
            <controlFilePrefix>sm-</controlFilePrefix>
            <controlFileNamePolicy>NNNNNNNN</controlFileNamePolicy>
            <controlFileSuffix>-control</controlFileSuffix>
          </groupInfoItem>
          <groupInfoItem>
            <groupId>voip</groupId>
            <controlFileDirectory>file://C:/IPDR/voip/</controlFileDirectory>
            <controlFilePrefix>voip_IT1_</controlFilePrefix>
            <controlFileNamePolicy>NNNNNN</controlFileNamePolicy>
            <controlFileSuffix>.log</controlFileSuffix>
          </groupInfoItem>
        </groupInfoList>
      </extension>
    </supportedProtocolItem>
  </supportedProtocolList>
</CapabilityRsp>
```

This capabilities file describes two subscription groups: vod and voip.

The voip subscription group has all its control files located in “<file://C:/IPDR/voip/>”. The names of the control files would all look like voip\_IT1\_000001.log. The following name would not be allowed voip\_IT1\_1.log since the control file name is required to have an eight digit sequence number embedded between the prefix and the suffix.

The sm (sm stands for Streaming Media) subscription group has all its control files located at “<http://www.best.com/sm>”. The names of the control-files would all look like sm-00000001-control.

## 4 Security Considerations

IPDR/FTP is dependent on the underlying security model maintained by the file system (or network shared file system) or other simple URL based file access schemes such as FTP, HTTP, or HTTPS. The use of FTP or HTTP in particular should not be used unless other policies protect access to the IPDR documents. FTP's basic authentication mechanism is a concern because passwords are passed in the clear. HTTP may or may not restrict access, in cases where authentication is used, mechanisms other than HTTP Basic authentication should be used to ensure that passwords or other credentials are not compromised.

In a user/group model, group level permission could be associated with specific document groups. This protocol does not currently provide any details related to the usage of file system permissions.