
Source: SA5 (Telecom Management)
Title: Rel-6 CR 32.343 File Transfer (FT) IRP CORBA SS
Document for: Approval
Agenda Item: 7.5.3

Doc1stLevel	Specific a	CR	R	Phase	Subject	Ca	VersCu	Doc2ndLev	Workitemsl D
SP-040800	32.343	001	--	Rel-6	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields - Align IDL style in FT IRP CORBA SS with IDL Style Guide in 32.150	F	6.0.0	S5-047032	OAM-NIM

CHANGE REQUEST

⌘ **32.343 CR 001** ⌘ rev **-** ⌘ Current version: **6.0.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: UICC apps ME Radio Access Network Core Network

Title:	⌘	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields - Align IDL style in FT IRP CORBA SS with IDL Style Guide in 32.150
Source:	⌘	SA5 (edwin.tse@ericsson.com)
Work item code:	⌘	OAM-NIM
		Date: ⌘ 19/11/2004
Category:	⌘	F
		Use <u>one</u> of the following categories:
		F (correction)
		A (corresponds to a correction in an earlier release)
		B (addition of feature),
		C (functional modification of feature)
		D (editorial modification)
		Detailed explanations of the above categories can be found in 3GPP TR 21.900 .
		Release: ⌘ Rel-6
		Use <u>one</u> of the following releases:
		2 (GSM Phase 2)
		R96 (Release 1996)
		R97 (Release 1997)
		R98 (Release 1998)
		R99 (Release 1999)
		Rel-4 (Release 4)
		Rel-5 (Release 5)
		Rel-6 (Release 6)

Reason for change:	⌘	Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields). Add the mandatory exception for optional method file_download_indication. Align IDL style with IDL Style Guide.
Summary of change:	⌘	Place IS-defined non-filterable parameters into remaining_body of CORBA structured event. Add the mandatory exception OperationNotSupported for optional method file_download_indication. Align IDL style with IDL Style Guide.
Consequences if not approved:	⌘	IRPAgent wastes CPU cycles on non-filterable parameters before emission of notification. EPIRP that does not support the optional method file_download_indication could not respond in a standard way. EPIRP IRPManager, on reception of non-standard response could be confused and could not conclude if the file_download_indication method is supported or not (e.g. temporarily malfunction of EPIRP).

Clauses affected:	⌘	5.2, Table 7, Table 8, A1, A2, A3								
Other specs affected:	⌘	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px;">Y</td> <td style="width: 20px;">N</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications ⌘ Test specifications ⌘ O&M Specifications ⌘	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Y	N									
<input type="checkbox"/>	<input checked="" type="checkbox"/>									
<input type="checkbox"/>	<input checked="" type="checkbox"/>									
<input type="checkbox"/>	<input checked="" type="checkbox"/>									
Other comments:	⌘									

Change in Clause 5.2

5.2 Operation parameter mapping

The FileTransferIRP: IS 3GPP TS 32.342 [7] defines semantics of parameters carried in operations across the FileTransferIRP. The following tables indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table 2: Mapping from IS listAvailableFiles parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
managementDataType	const short FileTransferIRPConstDefs::ManagementDataTypeType managementDataType	M
beginTime	FileTransferIRPConstDefs:: BeginTimeType UTCTimeType beginTime	M
endTime	FileTransferIRPConstDefs:: EndTimeType UTCTimeType endTime	M
fileInfoList	Return value of type FileTransferIRPConstDefs::FileInfoListType fileInfoList	M
status	FileTransferIRPConstDefs::ResultType Exception: ListAvailableFiles, InvalidTimes	M

Table 3: Mapping from IS fileDownloadIndication parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
fileInfoList	FileTransferIRPConstDefs::FileInfoListType fileInfoList	M
status	FileTransferIRPConstDefs::ResultType Exception: FileDownloadIndication, InvalidFileInfoList, ManagedGenericIRPSystem:OperationNotSupported	M

End of change in Clause 5.2

Change in Table 7, Table 8

Table 7: Mapping for notifyFileReady

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS attribute.	domain_name	M	It carries the IRP document version number string. See subclause 3.1. It indicates the syntax and semantics of the Structured Event as defined by the present document.
notificationType	type_name	M	This is the ET_FILE_READY of module of FileTransferIRPNotifDefs.
There is no corresponding IS attribute	event_name	M	It carries no information.
There is no corresponding IS attribute.	Variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
notificationId	One NV pair of <u>remaining_body</u> filterable_body_fields	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
fileInfoList	One NV pair of <u>remaining_body</u> filterable_body_fields	M	Name of NV pair is the FILE_INFO_LIST of module FileTransferIRPNotifDefs::notifyFileReady. Value of NV pair is FileInfoListType of module FileTransferIRPConstDefs.
additionalText	One NV pair of <u>remaining_body</u> filterable_body_fields	O	Name of NV pair is the ADDITIONAL_TEXT of module FileTransferIRPNotifDefs::notifyFileReady. Value of NV pair is a string.
There is no corresponding IS attribute.	remaining_body		

Table 8: Mapping for notifyFilePreparationError

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS attribute.	domain_name	M	It carries the IRP document version number string. See subclause 3.1. It indicates the syntax and semantics of the Structured Event as defined by the present document.
notificationType	Type_name	M	This is the ET_FILE_PREPARATION_ERROR of module of FileTransferIRPNotifDefs.
There is no corresponding IS attribute	event_name	M	It carries no information.
There is no corresponding IS attribute.	Variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
notificationId	One NV pair of remaining_body filterable_body_fields	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
fileInfoList	One NV pair of remaining_body filterable_body_fields	M	Name of NV pair is the FILE_INFO_LIST of module FileTransferIRPNotifDefs::notifyFilePreparationError . Value of NV pair is FileInfoListType of module FileTransferIRPConstDefs.
reason	One NV pair of remaining_body filterable_body_fields	M	Name of NV pair is the REASON of module FileTransferIRPNotifDefs::notifyFilePreparationError . Value of NV pair is a string.
additionalText	One NV pair of remaining_body filterable_body_fields	O	Name of NV pair is the ADDITIONAL_TEXT of module FileTransferIRPNotifDefs:: notifyFilePreparationError . Value of NV pair is a string.
There is no corresponding IS attribute.	remaining_body		

End of change in Table7, Table 8

A.1 IDL specification (file name "FileTransferIRPCConstDefs.idl")

```

// File: FileTransferIRPCConstDefs.idl

#ifndef FileTransferIRPCConstDefs_idl_FILETRANSFERIRPCONSTDEFS_IDL
#define FileTransferIRPCConstDefs_idl_FILETRANSFERIRPCONSTDEFS_IDL

#include "TimeBase.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: FileTransferIRPCConstDefs
This module contains commonly used definitions for FileTransferIRP.
=====
*/
module FileTransferIRPCConstDefs
{
    enum ResultType {OK, Failure};

    typedef TimeBase::UtcT UTCTimeType;
typedef UTCTimeType BeginTimeType;
typedef UTCTimeType EndTimeType;

    enum LocationChoice {Directory, URL};

    //The FileLocationType may be a directory path or a URL
    union FileLocationType switch (LocationChoice)
    {
        case Directory: string fileLocationDirectory;
        /* e.g. \\202.112.101.1\D:\user\performanceFiles\ */
        case URL: string fileLocationURL;
        /* e.g. ftp://nms.telecom_org.com/datastore/<fileName> */
    };

    typedef unsigned long FileSizeType; //the unit is byte
    typedef string FileCompressionType;
    typedef string FileFormatType;

    struct FileInfoType
    {
        FileLocationType fileLocation;
        FileSizeType fileSize;
        UTCTimeType fileReadyTime;
        UTCTimeType fileExpirationTime;
        FileCompressionType fileCompression;
        FileFormatType fileFormat;
    };

    typedef sequence<FileInfoType> FileInfoListType;

enum ManagementDataTypeType {PM, CM, IM, TE, CT, NL, CG, OT};
    const short PMManagementDataType = 1;//Performance Management
    const short CMManagementDataType = 2;//Configuration Management
    const short IMManagementDataType = 3;//Inventory Management
    const short TMManagementDataType = 4;//Test Management
    const short CTManagementDataType = 5;//Subscriber & Equipment Trace
    const short NLManagementDataType = 6;//Notification Log
    const short CGManagementDataType = 7;//Charging
    const short OTManagementDataType = 8;//Other Types

    /*
    Define the parameters specified in the notifyFileReady
    and notifyFilePreparationError notification.
    */
    interface AttributeNameValue

```

```

{
    const string FILE_INFO_LIST = "FILE_INFO_LIST";
    const string ADDITIONAL_TEXT = "ADDITIONAL_TEXT";
    const string REASON = "REASON";
};

};

#endif FILETRANSFERIRPCONSTDEFS_IDL

```

A.2 IDL specification (file name "FileTransferIRPSystem.idl")

```

// File: FileTransferIRPSystem.idl

#ifndef FILETRANSFERIRPSYSTEM_IDL FileTransferIRPSystem_idl
#define FILETRANSFERIRPSYSTEM_IDL FileTransferIRPSystem_idl

#include "ManagedGenericIRPCConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"
#include "FileTransferIRPCConstDefs.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: FileTransferIRPSystem
This module implements capabilities of FileTransferIRP.
=====
*/
module FileTransferIRPSystem
{
    exception InvalidTimes { string reason; };
    exception InvalidFileInfoList { string reason; };

    /*
    System fails to complete the operation. System can provide reason
    to qualify the exception. The semantics carried in reason
    is outside the scope of this IRP.
    */
    exception ListAvailableFiles { string reason; };
    exception FileDownloadIndication { string reason; };
    exception GetFileTransferIRPVersions { string reason; };
    exception GetFileTransferIRPOperationsProfile { string reason; };
    exception GetFileTransferIRPNotificationsProfile { string reason; };

    /*
    */
    interface FileTransferIRP
    {
        /*
        * IRPManager invoke this operation to get the files information
        */
        FileTransferIRPCConstDefs::ResultType list_available_files(
            in FileTransferIRPCConstDefs::ManagementDataTypeType short managementDataType,
            in FileTransferIRPCConstDefs::BeginTimeTypeUTCTimeType beginTime,
            in FileTransferIRPCConstDefs::EndTimeTypeUTCTimeType endTime,
            out FileTransferIRPCConstDefs::FileInfoListType fileInfoList
        )
        raises (ListAvailableFiles, InvalidTimes,
            ManagedGenericIRPSystem::InvalidParameter);

        /*
        * IRPManager invoke this operation to indicate completion of downloading files to IRPAgent
        */
        FileTransferIRPCConstDefs::ResultType file_download_indication(
            in FileTransferIRPCConstDefs::FileInfoListType fileInfoList
        )
        raises (FileDownloadIndication, InvalidFileInfoList,
            ManagedGenericIRPSystem::OperationNotSupported);
    }
}

```

```

/**
 * Return the list of all supported FileTransferIRP versions.
 */
ManagedGenericIRPConstDefs::VersionNumberSet get_FileTransfer_IRP_versions (
)
raises (GetFileTransferIRPVersions);

/**
 * Return the list of all supported operations and their supported
 * parameters for a specific FileTransferIRP version.
 */
ManagedGenericIRPConstDefs::MethodList get_FileTransfer_IRP_operations_profile (
    in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
)
raises (GetFileTransferIRPOperationsProfile,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/**
 * Return the list of all supported notifications and their supported
 * parameters for a specific FileTransferIRP version.
 */
ManagedGenericIRPConstDefs::MethodList get_FileTransfer_IRP_notifications_profile
(
    in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
)
raises (GetFileTransferIRPNotificationsProfile,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);
};

};

| #endif __FILETRANSFERIRPSYSTEM_IDL__

```

A.3 IDL specification (file name "FileTransferIRPNotifications.idl")

```

| // File: FileTransferIRPNotifications.idl
| #ifndef __FILETRANSFERIRPNOTIFICATIONS_IDL__
| #define __FILETRANSFERIRPNOTIFICATIONS_IDL__ __FILETRANSFERIRPNOTIFICATIONS_IDL__
|
| #include "FileTransferIRPConstDefs.idl"
| #include "NotificationIRPConstDefs.idl"
| #include "NotificationIRPNotifications.idl"
|
| // This statement must appear after all include statements
| #pragma prefix "3gppsa5.org"
|
| /* ## Module: FileTransferIRPNotifDefs
| This contains the specification of notifications of FileTransferIRP.
| =====
| */
| module FileTransferIRPNotifications
| {
|
|     /**
|     * Constant definitions for the FileReady notification
|     */
|     interface notifyFileReady: NotificationIRPNotifications::Notify
|     {
|         const string EVENT_TYPE = "notifyFileReady";
|
|         /**
|         * This constant defines the name of the fileInfoList property,
|         * which is transported in the filterable_body fields.
|         */
|     }
| }

```



```

* The data type for the value of this property is
* FileTransferIRPConstDefs::FileInfoListType.
*/
const string FILE_INFO_LIST =
    FileTransferIRPConstDefs::AttributeNameValue::FILE_INFO_LIST;

/**
* This constant defines the name of the additionalText property,
* which is transported in the filterable_body fields.
* The data type for the value of this property is string.
*/
const string ADDITIONAL_TEXT =
    FileTransferIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT;
};

/**
* Constant definitions for the FilePreparationError notification
*/
interface notifyFilePreparationError:
    NotificationIRPNotifications::Notify
{
    const string EVENT_TYPE = "notifyFilePreparationError";

    /**
    * This constant defines the name of the fileInfoList property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is
    * FileTransferIRPConstDefs::FileInfoListType.
    */
    const string FILE_INFO_LIST =
        FileTransferIRPConstDefs::AttributeNameValue::FILE_INFO_LIST;

    /**
    * This constant defines the name of the reason property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is string.
    */
    const string REASON =
        FileTransferIRPConstDefs::AttributeNameValue::REASON;

    /**
    * This constant defines the name of the additionalText property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is string.
    */
    const string ADDITIONAL_TEXT =
        FileTransferIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT;
};
};
| #endif FILETRANSFERIRPNOTIFICATIONS_IDL_FILETRANSFERIRPNOTIFICATIONS_IDL_

```

End of change in A1,A2,A3
