



Performance from Experience

Telcordia™ FOMS/FUSA Requirements for an Interface to a REMS

Release 9.0

Telcordia Technologies System Documentation
BD-FOMS-REQ-34
Issue 2
December 2003

ER Number: ZD-02143-01

Telcordia Technologies, Inc. Confidential — Restricted Access
This document and the confidential information it contains shall be distributed, routed or made available solely to authorized persons having a need to know within Telcordia, except with written permission of Telcordia.

Telcordia™ FOMS/FUSA Requirements for an Interface to a REMS Release 9.0

Prepared for Telcordia Technologies by:

Central Office Systems Development and Release Management

Technical contact:

Barbara L. Cruse
732 699-7856
bcruse@telcordia.com

Copyright © 2003 Telcordia Technologies, Inc. All rights reserved.

QMO Template Release 1.0.1 (09/2001).

Trademark Acknowledgments

Telcordia is a trademark of Telcordia Technologies, Inc.
SWITCH is a registered trademark of Telcordia Technologies, Inc.

All other brand or product names are trademarks of their respective companies or organizations.

Change History

Issue	Date	Revised By	Description of Change	Page(s)
2	Dec. 2003	B. L. Cruse	<ul style="list-style-type: none">• Requirements 3, 10, 19 and 21 clarified.	
			<ul style="list-style-type: none">• Requirement 22 added.	
			<ul style="list-style-type: none">• Requirement 13.10 added.	
			<ul style="list-style-type: none">• Section 11, fourth bullet modified.	

Contents

1 Introduction	7
1.1 Document Purpose and Scope	7
1.2 Target Audience	7
1.3 Project Overview	7
1.4 Review/Concurrence Information	7
1.5 Traceability	8
1.6 Disclaimer	8
1.7 Acronyms, Abbreviations, and Special Terms	9
2 General Information	10
2.1 Background	10
2.2 Challenge/Opportunity	10
2.3 Feature/Product Description	10
2.4 Benefit/Value	10
2.5 Security Risk Assessment	11
3 Processing/Flows	12
3.1 Current	12
3.2 Future	12
4 System Architecture	14
5 Detailed Requirements	15
5.1 Solution Data Requirements	15
5.2 Data Base Requirements	15
5.3 Conversion and/or Release Transition Requirements	16
5.4 Contract Requirements	17
5.5 FOMS Requirements	17
5.6 FUSA Requirements	22
6 System Interfaces/Impacts	23
7 Communication Requirements	24
8 Global Requirements	25
8.1 Availability/Reliability Requirements	25
8.2 Performance Requirements/Impacts	25
8.3 Security Requirements	25
8.4 Customer- or Site-Specific Requirements	25
8.5 Learning Support Requirements	26
9 Testing Considerations	27
10 Summary	28
11 Assumptions, Dependencies, Constraints	29
12 References	30
Appendix A: FCIF SECTION DEFINITION	31
A.1 Control Header Section Definition	32
A.2 Order Section Specification	33
A.3 FOMS Request/Response Section Specification	35

A.4 Sample Contracts 39

 A.4.1 FOMSTC Contract for SO with Related LST 39

 A.4.2 FOMSTC Example for Complex Circuit 42

 A.4.3 FOMSRR Request Contract for Frame Completion of a Circuit 44

 A.4.4 FOMSRR Response Contract for Frame Completion of a Circuit 44

 A.4.5 FOMSRR Request Contract for Frame Completion of A Multiline Order 45

 A.4.6 FOMSRR Request Contract for Issuing a Jeopardy 45

 A.4.7 FOMSRR Response Contract for Issuing a Jeopardy 46

 A.4.8 FOMSRR Request Contract for Running SCC 46

Appendix B: Glossary 47

1 Introduction

1.1 Document Purpose and Scope

The purpose of this document is to provide detailed requirements for enhancements to support the FOMS Interface to a REMS feature. The purpose of this feature is to provide an FCIF contract interface between FOMS/FUSA and a Robotic Element Management System which in turn interfaces with a robotic frame or connecting device.

ER Number: ZD-02143-01

This feature is targeted for the FOMS/FUSA Release 9.0 (11/03).

1.2 Target Audience

Audiences for this document include: Telcordia clients, system engineering, development, system test, performance, product management, installation, deployment, and learning support groups.

1.3 Project Overview

In conjunction with the Telcordia™ SWITCH® System, FOMS/FUSA provides wiring instructions for the frame for service and work orders and access to the SWITCH System database for the frame, the RCMAC, and other secondary users with asynchronous terminals.

The SWITCH System sends order data to FOMS in FCIF (Flexible Computer Interface Format). FOMS analyzes the order and decides if there is frame work to be done, or if the order will be suppressed. Similar logic is employed for correction passes and cancellations. Various inquiries and reports are provided for the convenience of the users.

FUSA transactions accept user input and create contracts to be sent to the SWITCH System to obtain inquiry or report data or to update an order or a circuit in the SWITCH System database.

FOMS also interfaces with Telcordia WFA/DI (Work and Force Administration/Dispatch In) and Telcordia Force via FCIF contract interfaces.

1.4 Review/Concurrence Information

These requirements were reviewed with Telcordia systems engineers, developers, and testers from FOMS/FUSA on December 9, 2003 and all blocking issues have been closed. All change requests for modifications to baselined documents are managed via the SWITCH System change control process.

1.5 Traceability

Detailed requirements in this document follow a numbering methodology for tracing the requirements back from development and test. Requirements are numbered according to the following format:

REMSINT-FOMS-R9.0-NNNN.nnn, where:

- **REMSINT** is the abbreviated identifier for the FOMS Interface to a REMS feature.
- **FOMS** is the unique identifier for the **FOMS/FUSA** product.

The “REMSINT-FOMS-” portion of the requirements numbering format is the prefix for all requirements found in this document for all external reference purposes. This prefix will be included in each requirement number.

- R9.0 is the release number in which the requirement was introduced.
- NNNN.nnn is a numeric value to distinguish the requirements within the document. “NNNN” represents a requirement and will range from 1-9999. “nnn” represents the optional sub-requirement number, which will range from 1-999.
- If it becomes necessary to add new requirements out of the original sequence after this document has been baselined, the original requirement numbers will have been frozen, so the first new requirement will be assigned the next highest number in sequence, but the requirement will be inserted where appropriate. Note that this means that after an update to this document, requirements may not appear in the document in ascending numerical order.
- If an original requirement is deleted after the document has been baselined, the content of the original requirement will be preserved by overstriking the text (as shown here: ~~deleted text~~).
- If an original requirement is changed after the document has been baselined, change bars will be used to mark the change, and a description of the change will be added to the **Change History** table.

1.6 Disclaimer

The information in this document represents Telcordia's current understanding of the functionality required to support the FOMS Interface to a REMS feature. This document is not a commitment on the part of Telcordia to implement this functionality. Such commitments are only made through the funding process.

Changes may be required during the implementation process due to unforeseen problems or other issues. Such changes may result in differences between these requirements and the way this feature is ultimately implemented. Significant changes will be communicated to the appropriate vendor and customer representatives.

1.7 Acronyms, Abbreviations, and Special Terms

ACT — Assignment Change Ticket
ATR — Area Transfer
CHP — Change Parameter
CIO — Company Initiated Order
CPT — Cable Pair Transfer
CTR — Channel Transfer
DIP — Dedicated Inside Plant
DTR — Dial Transfer
FOMS — Frame Operations Management System
FTR — Frame Transfer
FUSA — Frame User assignment System Access
IDP — In DIP
IJR — Initiate Jeopardy Reason
JAM — Jumper Activity Management
LST — Line and Station Transfer
MCT — Maintenance Change Ticket
MDF — Main Distributing Frame
PASM — Permanent Assembly
RDP — Reuse DIP
REMS — Robotic Element Management System
SCC — Send Connection Contract
SCM — Simple Completion by Mainframe
SET — Switchport Equipment Transfer
TCP/IP — Transmission Control Protocol/Internet Protocol
TOP — Transaction-Oriented Protocol
TOPCOM — Transaction-Oriented Protocol Communications
WAO — Wire Assembly Order
WFA/DI — Work Force Administration/Dispatch In
WO/LST — Work Order/Line and Station Transfer
WMC/APP — Work Manager Consolidation/Application-to-Application

2 General Information

2.1 Background

Historically telephone subscribers were given service by having a frame attendant or a central office technician running a wire from the F1 cable pair to a piece of switching equipment on a large metal structure known as a Main Distributing Frame. In recent years a new type of device has come on the market, one which is much smaller and involves no wiring but uses a robotic arm to connect the termination points of OEs and CPs based on instructions from a software system. The device is known as a robotic frame.

2.2 Challenge/Opportunity

Currently several clients have a robotic frame in a central office which is administered by the SWITCH System and FOMS. Connection information is input to the Element Management System (EMS) for the robotic frame in a manual mode. A frame technician prints the frame output from FOMS, and types the facility IDs into REMS, which translates the IDs into ports on the device and instructs the robot to make the connections.

Robotic frame vendors have requested Telcordia Technologies to develop an app-to-app interface between FOMS and a REMS to automatically transmit connection instructions and to receive frame completions and jeopardies from the REMS without human intervention.

2.3 Feature/Product Description

This feature provides a FOMS interface to a Robotic frame Element Management System (REMS). FOMS will prepare and send a contract to a REMS at an appropriate time based on a user-tunable table or when a COR pass comes in, for service orders, MCTs, SETs, WO/LSTs, CIOs and WAOs which require framework. The contract will contain the information necessary for the robotic device to make the proper connections or disconnections required by the order. The program which initiates the contract may be executed by the CRON utility program, by a frame user at a terminal, or via a FOMSRR contract from the REMS.

The REMS would send frame completions and jeopardies to FOMS via the standard FOMS request/response contract interface.

2.4 Benefit/Value

This feature will allow connection instructions to flow automatically from FOMS/FUSA to a REMS, and the required connections to be made or disconnected without

human intervention. This benefit is particularly valuable in remote offices where making a connection requires dispatching a technician to travel there and back.

2.5 Security Risk Assessment

No additional security risk is introduced as a result of this feature. Security for this interface is handled with standard security procedures.

3 Processing/Flows

3.1 Current

In an office with a robotic device serving as the connection point, the FOMS frame output is printed by a frame technician at the FOMS terminal as usual. It is then brought to the PC running the REMS and the facility IDs are typed into the system by the technician. The REMS instructs the device to make the connections and, when that is done, the frame technician goes back to the FOMS terminal and enters the frame completion.

3.2 Future

With the FOMS to REMS interface described in this document the frame technician disappears from the loop except for large transfers and those orders that fallout from the REMS due to complexity of circuit or mechanical failure. The connection/disconnection instructions will flow from FOMS to the REMS automatically and the frame completion status will flow from the REMS back to FOMS. Orders requiring coordination will be held in a queue in the REMS until released manually by a frame technician.

The following scenarios indicate that an order/circuit requires coordination:

1. A change order with both inward and outward activity; i.e., OT = CH and ACT=C,
2. Service order with embedded LST; LST falls under category 1 but entire order should be coordinated because the inward circuit depends on the LST so FOMS will send LST=Y flag with such circuits,
3. Order/circuit has FDT.

There are other scenarios that may be specific to a particular ILEC, for example:

- LNP-Port Out order; i.e., OC=LNP and TN going out,
- ADSL order; e.g., circuit contains TRE beginning with ADSL or something similar to that,
- Other order categories, such as UNE (Unbundled Network Element).

Due to a variety of local methods and procedures coordination of orders in the REMS should be controlled by a user-tunable table.

In a mixed office, that is, one with an MDF and a REMS device both serving as connection points, a frame technician would still be required to wire the connections on the MDF and manually enter the frame completion status into FOMS.

4 System Architecture

There are no changes to the FOMS/FUSA system architecture as a result of this feature.

5 Detailed Requirements

5.1 Solution Data Requirements

Not applicable.

5.2 Data Base Requirements

<REMSINT-FOMS-R9.0-1> There must be a tunability table entry (RSFREMS) to turn on this feature. Transaction **udinit** must be updated to accept the new entry.

<REMSINT-FOMS-R9.0-2> There must be a new field defined in the circuit level records to indicate that this circuit has or does not have REMS involvement.

<REMSINT-FOMS-R9.0-3> There must be a new client-tunable table (hereafter referred to as the FOMSTC table) which will specify when contracts may be sent to the REMS. The table will contain a date or date interval (and possibly a time of day) on which to send a contract based on the order type, action type, order category, related order type and FDD or DD of an order. Either FDD INT or DD INT may be specified, but not both for the same entry. The table should be set up so that the more specific entries precede the more general ones. It is strongly suggested that the last entry be generic to catch circuits which do not match any other entry. If the time of day is blank, the order may be sent anytime that day or on any subsequent day. If there is a time in the field, the order may be sent at or later than that time on the first day of the interval, and at any time on any subsequent day of the interval. Similar logic applies to the FDD and DD interval. A sample table follows.

OT	ACT	OC	REL OT	FDD INT	DD INT	PT
NC, T, CH	IN	LS,UNE			0	8:00
NC, T, CH	IN		CD, F, CH		0	10:00
NC, T, CH	IN			-2		14:00
CD,F,CH	OUT		NC, T, CH		0	6:00
CD,F,CH	OUT				+1	6:00
SET, WAO					-1	13:00
ETC.						
					0	9:00

<REMSINT-FOMS-R9.0-4> There must be a new **chp** parameter, **rems_frame**, which will contain the ID of any frames in the office which are REMS devices. This parameter must be taken into consideration when the interface is turned on, and

FOMS is preparing a contract to go to the REMS. Only connections/disconnections/reused facilities which are located on a frame which is a REMS device must be included in the contract. The format for the frame IDs is 'fxx,fxx' where xx are any 2 alphanumerics.

<REMSINT-FOMS-R9.0-5> There must be a new **chp** parameter, **rems_mixed**, to indicate that this is a 'mixed' office. Mixed means that there is both a REMS device and a subscriber MDF in use in this office. A value of 'Y' means it is a mixed office; a value of 'N' means it is not. Anytime the FOMS software determines that the REMS Interface is turned on for a wire center, it must then check for the **rems_mixed** parameter to see if it is a mixed office. If the parameter does not exist, FOMS should assume a value of 'N'. If that parameter happens to be in the PM table, but the REMS interface is not turned on, the parameter will be ignored.

<REMSINT-FOMS-R9.0-21> There must be a **chp** parameter called **rems_app**, which consists of 2 alphanumerics, which will associate the wire center with a REMS application name.

<REMSINT-FOMS-R9.0-22> There must be a **chp** parameter called **rems_name**, which may be 1-8 alphanumerics, but may not begin with 'fo'.

5.3 Conversion and/or Release Transition Requirements

<REMSINT-FOMS-R9.0-6> To assist in the conversion of the REMS database and on-going synchronization due to Telephone number changes and large work order changes, Telcordia shall provide a SWITCH System Licensed Software database procedure that performs the following functions.

1. Input to the procedure will be Wire Center and Frame ID (e.g., "F01") and date. The program will find the frame and proceed by frame location or cable pair range to read through the database. For working circuits, the program will create a flat file in the following format.

2. The output will look like:

WC Frame ID

cable-pair ----OE (IC dependent)----TN (format NPA NXX-NNNN)/CKID/blank

cable-pair----OE----TN

cable-pair----OE----TN

3. In the event of large pending work order, the program should output the future view of the circuit as of the parameter date provided as input.
4. The output dataset will be named by JCL parameters and be sent to the REMS system using FTP (file transfer protocol) or such electronic or physical media as the SWITCH System clients select.

This feature does not include a FOMS conversion or transition program. Responsibility for initializing the REMS database with the facility IDs belongs to the REMS vendor. All orders that have been printed should be wired and frame completed when the interface is turned on. Unprinted orders in FOMS which involve the REMS device will be sent to the REMS at the appropriate time after the interface is turned on.

5.4 Contract Requirements

<REMSINT-FOMS-R9.0-7> When a decision has been made to send connection data for an order to the REMS, it is sent via a FOMSTC contract. The FOMSTC contract is sent with a standard *C1 header, and always contains a *ORD section. See Appendices A and B for details of the FOMSTC contract.

<REMSINT-FOMS-R9.0-8> The existing FOMSRR contract will be used by the REMS to send frame completions and jeopardies to FOMS. It will be sent with a standard *C1 header. FOMS will use the FOMSRR contract to send a response to the REMS in case of an error, and to send a successful response if the REMS wants one. See Appendices A and B for details of the FOMSRR contract.

5.5 FOMS Requirements

<REMSINT-FOMS-R9.0-9> The requirements which follow apply to Service Orders, MCTs, SETs, WO/LSTs, CIOs and WAOs. CPTs, DTRs, ATRs, FTRs, JAMs and CTRs will not be sent to the REMS. Any order/circuit which is determined to be suppressed in FOMS will not be sent to the REMS. Initial passes of service orders, SETs, WO/LSTs, CIOs and WAOs, or a subsequent pass if the initial pass was not yet sent, will be sent via the **scc** program. After one pass has been sent, subsequent passes (i.e., modifications and cancellations) as well as MCTs will be sent by FOMS upon establishment.

<REMSINT-FOMS-R9.0-10> A program named **scc**, Send Connection Contracts, is needed to be run via the CRON utility, manually from a terminal, or via a FOMSRR contract from the REMS. It will not require any input but a period, but will accept an order number and a circuit ID. If an order number (ORD) or order number and circuit ID is input, it will make sure that the order type is one of the valid types for this interface and that the order is REMS-involved, send that order/circuit to the REMS and not read the FOMSTC table. The requested order/circuit data should be sent to the REMS regardless of the value of the status or print tag.

If order number is not input, it will perform the following functions. Go through the order file, looking for service orders, MCTs, SETs, WO/LSTs, WAOs and CIOs which have REMS involvement, but have not been sent. See if that order/circuit qualifies to be sent at the present time, by going through the table line by line looking for a match on OT, ACT, OC and REL OT. If a match is found, then

the PT and FDD or DD INT should be checked, and a decision is made to send or not send the order/circuit at this time. In either case the search through the table entries stops at this point. Once the decision has been made to send an order to the REMS, the IN, OUT and REU jumpers must be determined for the circuit and saved until all circuits in the order are analyzed. When the entire order has been checked, a FOMSTC contract should be prepared for that order and sent to the REMS. The circuit record for each circuit included in the FOMSTC contract must be marked as 'sent to the REMS' and the print tag must be set to 'printed', if the circuit is not 'mixed'.

<REMSINT-FOMS-R9.0-11> Transaction **scm** must

- <R.11.1> accept input via a FOMSRR contract from the REMS,
- <R.11.2> only allow input from a terminal with 'OVR Y' if the FOMS-to-REMS interface is on, but `rems_mixed` is not set to 'y',
- <R.11.3> allow input from a FOMSRR contract and from a terminal (without OVR Y) if `rems_mixed` is set to 'y', and keep track of both completions; if OP ALL is input, only give frame completion to the applicable circuits,
- <R.11.4> not give a status of frame completion to a circuit in a mixed office until both completions have been done,
- <R.11.5> not require CAN Y when a cancelled circuit is being completed via a FOMSRR contract from the REMS,
- <R.11.6> not do any automatic completion of related orders when completing an order/circuit for the REMS; if there is a dependent related order blocking completion, it should error.
- <R.11.7> not error if it receives an unnecessary completion from the REMS.

<REMSINT-FOMS-R9.0-12> Transaction **ijr** must

- <R.12.1> accept input via a FOMSRR contract from the REMS,
- <R.12.2> only allow input from a terminal with 'OVR Y' if the REMS interface is on, but `rems_mixed` is not set to 'y',
- <R.12.3> allow input from a FOMSRR contract or from a terminal (without OVR Y) if `rems_mixed` is set to 'y'.

<REMSINT-FOMS-R9.0-13> A new software module is required to prepare the FOMSTC contract from the data in the FOMS database for an order. Some general observations about the contract follow.

- <R.13.1> Inward jumpers translate to IN aggregates; therefore a circuit with all inward jumpers will result in a CKT aggregate with only IN subaggregates.
- <R.13.2> Outward jumpers translate to OUT aggregates; therefore a circuit with all outward jumpers will result in a CKT aggregate with only OUT subaggregates.

- <R.13.3> If IDPs, RDPs and PASMs are not suppressed, they result in a CKT aggregate with a REU subaggregate.
- <R.13.4> The MP (Multipoint) tag will appear in an aggregate whenever the OE is connected to two facilities.
- <R.13.5> The LB (Loopback) tag will appear in an aggregate whenever a CP is connected to another CP.
- <R.13.6> If an order involves a working circuit, the CKT aggregate may contain any combination of IN, OUT and REU subaggregates. Reused jumpers in the circuit should be shown in a REU subaggregate.
- <R.13.7> A circuit with just TN and an OE which is going in or out should be sent to the REMS. The CKT aggregate will contain an IN or OUT subaggregate with just one FAC subaggregate for the OE.
- <R.13.8> If FOMS is sending a contract for an SET, the CKT aggregate should contain the out OE of the circuit instead of the TN or CKID, so that the REMS will use that for the SCM and it will execute successfully.
- <R.13.9> If FOMS is sending a contract for a WO/LST, the CKT aggregate should contain the out CP of the circuit instead of the TN or CKID, so that the REMS will use that for the SCM and it will execute successfully.
- <R.13.10> The *ORD.CKT.RELORD aggregate will be populated if the circuit being processed has a dependent related circuit which has not been frame completed.

<REMSINT-FOMS-R9.0-14> Cancellations:

- <R.14.1> If FOMS receives a cancellation of an order/circuit that has not yet been sent to the REMS, it goes to withdrawn status in the FOMS database and nothing is sent to the REMS regarding the order/circuit.
- <R.14.2> If FOMS receives a cancellation of an order/circuit which has been sent to the REMS and it is not frame complete, FOMS should update its database as usual, and send a FOMSTC contract with a *ORD.CKT aggregate with the appropriate IN or OUT aggregates to notify the REMS to undo the original work plus the NOSCM=CAN tag. The print tag should be set to '4' and the 'sent to the REMS' tag should be set to 'sent'. If the REMS has the original instructions for the cancelled order/circuit in its hold queue, it should delete them, send a FOMSRR/SCM contract to FOMS, and discard the cancellation instructions. If the REMS has already implemented the original instructions, it should implement the cancellation instructions, and send FOMS a FOMSRR/SCM contract.
- <R.14.3> If FOMS receives a cancellation of an order/circuit which has been sent to the REMS and is frame complete, FOMS should update its database as usual (including removing the frame completion), and send a FOMSTC contract with a *ORD.CKT aggregate with the appropriate IN or OUT aggregates to notify the REMS to undo the original work. The print tag should be set to '4' and the 'sent to the REMS' tag should be set to 'sent'. The

REMS will execute the cancellation instructions and send FOMS a FOMSRR/SCM contract.

<REMSINT-FOMS-R9.0-15> Modifications including ACTs and Change of Circuit ID only passes:

- <R.15.1> If FOMS receives a modification of an order/circuit which has not yet been sent to the REMS, FOMS treats the modification as if it were a first pass in the FOMS database and a FOMSTC contract will be sent to the REMS at the appropriate time.
- <R.15.2> If FOMS receives a modification of an order/circuit which has been sent to the REMS but is not frame complete, FOMS should update its database as usual (including resetting the print tag), and send the REMS a FOMSTC contract with the required IN, OUT and REU aggregates to notify the REMS to undo the original work and do the new work, and include the NOSCM=MOD tag. The print tags and 'sent to the REMS' tag should be set accordingly. The REMS will execute both the original and the new instructions.
- <R.15.3> If FOMS receives a modification of an order/circuit which is frame complete, FOMS should update its database as usual (including removing the frame completion), and send the REMS a FOMSTC contract with a *ORD.CKT aggregate with the required IN, OUT and REU aggregates to notify the REMS to undo the original work and do the new work. The print tags and the 'sent to the REMS' tag should be set accordingly. The REMS will execute the modification instructions and send FOMS a FOMSRR/SCM contract.
- <R.15.4> If FOMS receives a modification of an order/circuit which has been sent to the REMS and only the Circuit ID of the circuit is changing, FOMS should update its database as usual (but not reset the print tag) and send a FOMSTC contract to the REMS, whether the status is frame complete or not. It should contain both the old and new circuit ID and no IN, OUT or REU aggregates. The REMS should either update the Circuit ID in the instructions in the hold queue, or if the work has already been performed, update the Circuit ID in its database if necessary. A new FOMSRR/SCM contract is not required in the latter case.
- <R.15.5> If FOMS receives a modification of an order/circuit which has been sent to the REMS and the circuit ID and circuit data are changing, FOMS should update its database as usual and send a FOMSTC contract to the REMS, whether the status is frame complete or not. It should contain both the old and new circuit IDs and the appropriate IN, OUT and REU aggregates to undo the previous work and do the new work. If the order/circuit did not have frame completion in FOMS, the contract should contain NOSCM=MOD. If the contract contains the NOSCM tag, the REMS should look in its hold queue for a match on the old circuit ID, and either change the old circuit ID to the new one or at least make sure that both instructions are executed and the new circuit ID is the one stored in the REMS database.

If the NOSCM tag is not present, the REMS should just execute the new instructions and the new ID should be the one stored in the database.

<REMSINT-FOMS-R9.0-16> Change of Due Date/Frame Due Date:

- <R.16.1> If FOMS processes a Change of Due Date/Frame Due Date only pass for an order and the order has not yet been sent to the REMS, FOMS should update its database as usual, and as far as the REMS is concerned, it will be treated as a first pass.
- <R.16.2> If FOMS processes a Change of DD or FDD only pass for an order and the order or any circuit in the order has been sent to the REMS, but is not frame complete, FOMS should update its database as usual, and do one of the following: a) if the order/circuit is for strictly inward work (action tag = 2), no FOMSTC contract will be sent to inform the REMS of the change of DD/FDD whether the DD/FDD is moved in or out, or b) if the order/circuit is for outward work or both inward and outward work (FOMS action tag = 1, 3, 4, or 5), FOMS will send a FOMSTC contract with a NOSCM=CDD tag to undo the original work if the DD/FDD is moved out. The FOMSTC contract should contain the CDD=Y tag. If the DD/FDD is moved in, no contract need be sent.
- <R.16.3> If FOMS processes a Change of DD/FDD only pass for an order and the order/circuit is frame complete, FOMS should update its database as usual, and do one of the following: a) if the order/circuit is for strictly inward work (action tag = 2), no FOMSTC contract will be sent to inform the REMS of the change of DD/FDD whether the DD/FDD is moved in or out, or b) if the order/circuit is for outward work or both inward and outward work (action tag = 1, 3, 4, or 5), FOMS will send a FOMSTC contract to undo the original work if the DD/FDD is moved out. If the DD/FDD is moved in, no contract need be sent. The FOMSTC contract should contain the CDD=Y tag.
- <R.16.4> If FOMS processes a modification pass which consists of both a change of due date or frame due date and a change to the circuit, it should update its database as usual, and do one of the following: a) if the order/circuit has not been sent to the REMS, do not send a contract at this time, b) if the order has been sent to the REMS but is not frame complete, FOMS must send a FOMSTC contract to the REMS with the instructions to undo the previous work and do the new work. The FOMSTC contract will contain the new DD/FDD and NOSCM=MDD, but will not contain the CDD tag; if the original instructions have been executed, the REMS will execute the new ones but if the original instructions are in the hold queue, the REMS must link up the old and new instructions and execute them both when the order is released from the queue, c) if the order/circuit is frame complete, FOMS must send a FOMSTC contract to the REMS with the instructions to undo the previous work and do the new work. The FOMSTC contract will contain the new DD/FDD but will not contain the CDD tag.
- <R.16.5> If FOMS processes a modification pass which consists of a change of facility, a change of circuit ID and a change of DD or FDD, it should update

its database as usual, and do one of the following: a) if the order/circuit has not been sent to the REMS, do not send a contract at this time, b) if the order has been sent to the REMS, but is not frame complete, FOMS should send a FOMSTC contract to the REMS with the new DD/FDD, the old and new circuit IDs, and instructions to undo the previous work and do the new work and NOSCM=MDD; if the original instructions have been executed, the REMS will execute the new ones but if the original instructions are in the hold queue, the REMS must link up the old and new instructions (using the old circuit ID to find the original instructions) and execute them both when the order is released from the queue, c) if the order/circuit is frame complete, FOMS should send a FOMSTC contract to the REMS with the new DD/FDD, the old and new circuit IDs, and instructions to undo the previous work and do the new work. It will not contain a CDD tag.

<REMSINT-FOMS-R9.0-17> In a mixed office the REMS interface must work with either FOR or FWM and with either the interface to Telcordia™ Force or the interface to Telcordia™ WFA/DI turned on.

<REMSINT-FOMS-R9.0-18> In addition to the aforementioned instances of modifications, cancellations and changes of due date, Maintenance Change Tickets (MCTs) will be sent to the REMS upon establishment.

5.6 FUSA Requirements

There is no impact to FUSA transactions as a result of this feature.

6 System Interfaces/Impacts

This document describes a new interface from FOMS to the REMS. There are no impacts to any other interface that FOMS maintains.

7 Communication Requirements

<REMSINT-FOMS-R9.0-19> Messages that originate from or terminate at the REMS will use a TOP/TCP/IP network connection to send/receive FCIF contracts to/from FOMS. TOP is a Transaction-Oriented Protocol which provides an interface that allows applications to establish, accept, and terminate sessions with a foreign partner. TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of protocols that facilitate the interconnection of dissimilar computer systems across networks. The software to implement this will be TOPCOM. It uses a TCP/IP transport service to transport the TOP traffic between the two partners.

Messages that originate from or terminate at FOMS will use Work Manager Consolidation/Application-to-Application (WMC/APP) software to send/receive contracts to/from the REMS. WMC/APP then uses a TOP/TCP/IP connection to send contracts to and receive contracts from the REMS. WMC/APP is responsible for the safestore and queuing of the FCIF contracts for FOMS.

Messages will be routed based on the Source, Destination, and FOMS wire center (NPANXX) fields in the Control Header (*C1) section of the FCIF message.

The FOMS system will route to different REMS systems based upon the NPANXX. Similarly, the REMS system is required to manage multiple TOPCOM sessions and have the capability of routing to specific sessions (and therefore specific FOMS machines) based upon the NPANXX of the message. The interface necessary to address specific TOPCOM sessions is described in the TOPCOM documentation.

8 Global Requirements

8.1 Availability/Reliability Requirements

This feature will be available in FOMS/FUSA Release 9.0 (11/03).

8.2 Performance Requirements/Impacts

This feature should have minimal impact on FOMS/FUSA performance. The new transaction, **scc**, will be on the same order of magnitude as a **for** transaction in the same wire center. Since the work done by each individual transaction will not change much with transaction frequency, the total impact to FOMS/FUSA CPU consumption and performance will depend on the frequency and timing of these transactions. Other transactions such as **scm** will show a modest increase in CPU consumption, when the feature is active. There will also be an increase in processing required to send messages to the REMS in the event of cancellations or changes to orders that have already been sent, but the effect is expected to be minimal because of relatively low volume. There may be some savings of CPU due to the decreased need to print and manage the affected orders (**foi**, **for**, etc.). Overall there should be less than a 4% increase in CPU consumption for the affected wire centers, assuming a reasonable schedule for **scc**.

8.3 Security Requirements

This is an OSMINE feature and under the control of a feature key. See Requirement 1 for details.

8.4 Customer- or Site-Specific Requirements

Except for the feature key, there are no customer or site specific requirements.

8.5 Learning Support Requirements

<REMSINT-FOMS-R9.0-20> The following documents are impacted by this feature and must be updated for Release 9.0.

Document Number	Document Title	Area of Impact
BR 752-101-901	FOMS/FUSA Functional Product Specifications	New feature
BR 752-103-901	FOMS/FUSA System Description	Add Section 2.2.8 on new interface.
BR 752-105-910	FOMS/FUSA Installation Guide	Add information on how to set up CRON to run scc .
BR 752-106-920	FOMS/FUSA Application Administration and Transactions	udinit , chp parameters, setting up feature
BR 752-108-930	FOMS/FUSA Line User Guide	Add transaction scc
BR 752-110-910	FOMS/FUSA Database Design	Update to circuit level records

9 Testing Considerations

This feature requires pairwise testing between FOMS and the REMS.

10 Summary

This document describes a new interface from FOMS to a REMS, an element management system for a robotic device. Connection instructions are sent via the interface for service orders and some work orders. This will enable the orders to be 'wired' by the robotic device, and frame completion status to be sent back to FOMS by the REMS without the need for a frame technician to get involved.

11 Assumptions, Dependencies, Constraints

- This feature requires a corresponding release of the REMS.
- This feature includes WMC/APP and TOPCOM on the FOMS side.
- This feature includes TOPCOM or equivalent on the REMS side.
- This feature applies to service orders and the following subset of work orders: MCTs, SETs, WO/LSTs, CIOs and WAOs.
- This feature does not include a conversion/transition program to send orders already in FOMS to the REMS when the interface is turned on.
- Passes of orders/circuits without actual frame work will not be sent over this interface unless the Frame Suppression table is set to do so.
- The REMS system will meet the FOMSRR contract interface for sending frame completion notices and jeopardy data to FOMS.
- FOMS will only interface with one REMS system per wire center.
- Suppression rules as specified in the FX table apply to a switching entity, not to a particular frame.
- There is no difference between an ACT and a COR pass as far as the interface to the REMS is concerned, nor does it matter if the ACE-like Frame Output feature is on or not.
- The REMS will treat a REU aggregate like an IN aggregate: if the connection is already there, do nothing; but if it does not exist, make the connection.

12 References

Not applicable.

Appendix A: FCIF SECTION DEFINITION

Appendix A contains detailed definitions of each FCIF section (including the *C1 header) that can appear in FOMS to a REMS application-to-application interface contracts. The sections are:

*C1 Header

*FOMSTC

*FOMSRR

A.1 Control Header Section Definition

The Control Header (*C1) Section is a fixed fielded section of fixed length. The fixed length is 58 characters. The *C1 Section contains order-level control information. The *C1 Section is required in all contracts passed between FOMS and a REMS. The following defines the *C1 header for all FOMS/REMS interface contracts:

*C1= AAABBBCCCCCCCCCCCCDEEEFFFFFFFGGGGGGGGGHHHHHHHHHIIJJJJJK
LMNO;%

Field No.	FIELD	DESCRIPTION	DATA VALUES	NOTES
1	AAAAAA	contract name	FOMSTC FOMSRR	1
2	CCCCCCCCCCC C	order id	alphanumeric	7
3	D	correction suffix	alphanumeric	2
4	EEE	version number	alphanumeric	2
5	FFFFFF	FOMS wire center	numeric	3
6	GGGGGGGG	source of message	alphanumeric	4
7	HHHHHHHH	destination of message	alphanumeric	5
8	II	priority	numeric: -7 to +7, or blank	2
9	JJJJJ	due date	yymmdd or blank	2
10	K	due date change flag	Y blank	2
11	L	solicited/unsolicited flag	S U blank	2
12	M	status	blank	2
13	N	manual assistance flag	blank	2
14	O	message format	0	6

1. FOMS to REMS contracts are: FOMSTC and FOMSRR; the only REMS to FOMS contract is FOMSRR.
2. This field will be blank in the FOMS/REMS interface.
3. This field contains the NPANXX of the wire center.
4. This field is set to FOMS (left justified) for contracts going from FOMS to a REMS, and is set to REMS (left justified) for contracts going from a REMS to FOMS.
5. This field is set to REMS (left justified) for contracts going from FOMS to a REMS and is set to FOMS for contracts going from a REMS to FOMS.
6. 0 = FCIF
7. This field will be blank in the FOMSRR contract; it will be filled in by FOMS in the FOMSTC contract for testing purposes; it may be ignored by the REMS.

A.2 Order Section Specification

The Order (*ORD) section contains information about all circuits in an order that require data to be sent to the REMS.

The *ORD section is required for all FOMSTC contracts.

*ORD SECTION DESCRIPTION								APP.	NOTES
*ORD{								1	
	ORDNO							1	
	OT							1	
	CDD							0,1	
	DD							1	
	FDD							1	
	FDT							0,1	
	WC							1	
	PRI							1	
	CKT{							1+	
		CTTYPE						1	
		CTID						1	
		OCTTYPE						0,1	
		OCTID						0,1	
		ADSR						0,1	
		SSM						0,1	
		SSP						0,1	
		OC						0,1	
		ACT						1	
		LST						0,1	
		CLS						0,1	
		NOSCM						0,1	
		RELORD{						0+	
			ORDNO					1	
			CTTYPE					1	
			CTID					1	
			OT					1	
			ACT					1	
		}							
		IN{						0+	

*ORD SECTION DESCRIPTION								APP.	NOTES
			FAC{					1,2	
				TYPE				1	
				ID				1	
				FRLOC				1	
				SPECFUNC				0,1	
				MP				0,1	
				LB				0,1	
			}						
		}							
		OUT{						0+	
			FAC{					1,2	
				TYPE				1	
				ID				1	
				FRLOC				1	
				SPECFUNC				0,1	
				MP				0,1	
				LB				0,1	
			}						
		}							
		REU{						0+	
			FAC{					1,2	
				TYPE				1	
				ID				1	
				FRLOC				1	
				SPECFUNC				0,1	
				MP				0,1	
				LB				0,1	
			}						
		}							
	}								
}%									

A.3 FOMS Request/Response Section Specification

The FOMS Request/Response (*FOMSRR) section contains data requesting that a FOMS transaction be run, or the output of a FOMS transaction to be sent back to the requesting system.

The *FOMSRR section is required for all FOMSRR contracts.

*FOMSRR SECTION DESCRIPTION							APP.	NOTES
*FOMSRR{							1	
STDRQ{							0,1	13
	TRAN						1	
	HDATA{						1	1
		"TAG"					1+	5
	}							
	IDATA{						0,1	2
		"TAG"					1+	5
	}							
	ODATA{						0,1	3
		"TAG"					1+	5
	}							
	RDATA{						0,1	4
		"TAG"					1,3	5
}	}							
PLHDR{							1	
	FOMSRSP						0,1	
	LCLID						0,1	12
	FOMSDBG						0,1	6
	REDIRECT						0,1	7
}								
STDRS{							0,1	13
	NPANXX						0,1	
	TRAN						1	
	STATUS						1	8
	STDERR{						0+	9
		NUML					1	
		LINE					1,33	14
	}							
	STDOUT{						0+	10
		NUML					1	
		LINE					1,33	14
	}							
	DEBUG{						0+	11
		NUML					1	

*FOMSRR SECTION DESCRIPTION								APP.	NOTES
			LINE					1,33	14
		}							
}	%								

1. One HDATA aggregate contains all H-line data.
2. One IDATA aggregate contains all I-line data.
3. One ODATA aggregate contains all O-line data.
4. One RDATA aggregate contains all R-line data.
5. The tag name is the actual process input prefix and the tag value is the data for that prefix as it would appear on an H, I, O, or R line in a transaction. For example, an H-line of 'h ord n123' would appear in the HDATA aggregate as 'ORD=N123'. There may be multiple tags in an aggregate and in some cases there may be multiple occurrences of a given tag.
6. Appears if debug feature in the FOMS transaction is to be turned on.
7. Appears if debug output is to be redirected to a file.
8. STATUS=0 if totally successful; non-zero otherwise.
9. Appears if there is error output.
10. Appears if there is standard output.
11. Appears if debug output was not directed to a file.
12. Local ID; if sent by the REMS it will be echoed back by FOMS.
13. Either STDRQ or STDRS will be present, but not both.
14. Limit of 33 'LINE' tags per aggregate.

A.4 Sample Contracts

A.4.1 FOMSTC Contract for SO with Related LST

The following *ORD section is for a two-line order consisting of an inward circuit with a related LST to free up a cable pair.

*ORD EXAMPLE						
*ORD{						
ORDNO=C123;						
OT=CH;						
DD=20020521;						
FDD=20020521;						
WC=MA;						
PRI=2;						
	CKT{					
		CTTYPE=TN;				
		CTID=201-822-1698;				
		ACT=C;				
		CLS=R;				
		LST=Y;				
		IN{				
			FAC{			
				TYPE=OE;		
				ID=100-201-305;		
				FRLOC=F02-001;		
			}			
			FAC{			
				TYPE=CP;		
				ID=9-213;		
				FRLOC=F02-001;		
			}			
		}				
		OUT{				
			FAC{			
				TYPE=OE;		
				ID=100-201-305;		

*ORD EXAMPLE (Continued)							
					FRLOC=F02-001;		
				}			
				FAC{			
					TYPE=CP;		
					ID=9-463;		
					FRLOC=F02-001;		
				}			
			}				
		}					
		CKT{					
			CLS=R;				
			CTTYPE=TN;				
			CTID=201-822-0524;				
			ACT=I;				
			RELORD{				
				ORDNO=C123;			
				CTTYPE=TN;			
				CTID=201-822-1698;			
				OT=CH;			
				ACT=C;			
			}				
			IN{				
				FAC{			
					TYPE=OE;		
					ID=102-301-212;		
					FRLOC=F02-001;		
				}			
				FAC{			
					TYPE=CP;		
					ID=9-463;		
					FRLOC=F02-001;		
				}			
			}				
		}					
}%							

A.4.2 FOMSTC Example for Complex Circuit

The following *ORD section is for a change order to a POTS circuit inserting a permanent assembly consisting of 2 MEs between the OE and CP.

*ORD EXAMPLE						
*ORD{						
ORDNO=C456;						
OT=CH;						
DD=20020621;						
FDD=20020621;						
WC=MA;						
PRI=2;						
	CKT{					
		CLS=B;				
		CTTYPE=TN;				
		CTID=201-321-0006;				
		ACT=C;				
		IN{				
			FAC{			
				TYPE=OE;		
				ID=100-013-002;		
				FRLOC=F10-001;		
				}		
			FAC{			
				TYPE=ME;		
				ID=SPLT.0001;		
				FRLOC=F10-001;		
				}		
		}				
		IN{				
			FAC{			
				TYPE=ME;		
				ID=MTPT.0001;		
				FRLOC=F10-001;		
				}		
			FAC{			

*ORD EXAMPLE (Continued)							
					TYPE=CP;		
					ID=50-0002;		
					FRLOC=F10-001;		
				}			
			OUT{				
				FAC{			
					TYPE=OE;		
					ID=100-013-002;		
					FRLOC=F10-001;		
				}			
				FAC{			
					TYPE=CP;		
					ID=50-0002;		
					FRLOC=F10-001;		
				}			
			}				
			REU{				
				FAC{			
					TYPE=ME;		
					ID=SPLT.0001		
					FRLOC=F10-001;		
				}			
				FAC{			
					TYPE=ME;		
					ID=MTPT.0001		
					FRLOC=F10-001;		
				}			
		}					
}							

A.4.3 FOMSRR Request Contract for Frame Completion of a Circuit

*FOMSRR SECTION EXAMPLE						
*FOMSRR{						
STDRQ{						
	TRAN=SCM;					
	HDATA{					
		ORD=N123;				
		TN=201-822-1698;				
		OP=RJR;				
}	}					
PLHDR{						
	LCLID=1234;					
}						
%						

A.4.4 FOMSRR Response Contract for Frame Completion of a Circuit

*FOMSRR SECTION EXAMPLE			
*FOMSRR{			
STDRS{			
	TRAN=SCM;		
	STATUS=0;		
	STDOUT{		
		NUML=4;	
		LINE=**PARTIAL COMPLETION;;	
		LINE= CKT-ID: CKID 201.822.1698;	
		LINE=** TRANSACTION PROCESSED(SCM) - COMPLETIONS: 1;	
		LINE=**SCM COMPLETED 05-06-02 10:57;	
	}		
}			
PLHDR{			
	LCLID=1234;		
}			
%			

A.4.5 FOMSRR Request Contract for Frame Completion of A Multiline Order

*FOMSRR SECTION EXAMPLE						
*FOMSRR{						
STDRQ{						
	TRAN=SCM;					
	HDATA{					
		ORD=N123;				
		OP=RJR;				
		OP=ALL;				
}	}					
%						

A.4.6 FOMSRR Request Contract for Issuing a Jeopardy

*FOMSRR SECTION EXAMPLE						
*FOMSRR{						
STDRQ{						
	TRAN=IJR;					
	HDATA{					
		ORD=N123;				
		TN=201-822-1698;				
		JR=A1;				
	}					
	RDATA{					
		RMK=EQP FAILURE;				
	}					
}						
%						

A.4.7 FOMSRR Response Contract for Issuing a Jeopardy

*FOMSRR SECTION EXAMPLE			
*FOMSRR{			
STDRS{			
	TRAN=IJR;		
	STATUS=0;		
	STDOUT{		
		NUML=4;	
		LINE=**ORDER N123 HAS BEEN GIVEN JEOPARDY	
		STATUS;	
		LINE=CKT-ID: CKID 201.822.1698;	
		LINE=** JEOPARDY REASON: A1 05-06-02 10:57;	
		LINE=** IJR COMPLETED 05-06-02 10:57;	
	}		
}			
PLHDR{			
	LCLID=1234;		
}			
%			

A.4.8 FOMSRR Request Contract for Running SCC

*FOMSRR SECTION EXAMPLE						
*FOMSRR{						
STDRQ						
{						
	TRAN=SCC;					
	HDATA{					
		ORD=N123;				
		CKID=279.822.6000.0001;				
	}					
}						
PLHDR						
{						
	LCLID=1234;					
}						
%						

Appendix B: Glossary

Appendix B contains detailed definitions of each FCIF tag that can appear in FOMS/REMS interface contracts. Following the heading “Appears in:”, the full path for each data item is shown using the notation CONTRACT.*SECTION.AGGREGATE(S).

Name: ACT

Full Name: Action Type

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: I | O | C

Appears in: FOMSTC.*ORD.CKT, FOMSTC.*ORD.CKT.RELORD

Description: This tag indicates the action type of the circuit or related circuit: I=IN, O=OUT, and C=CHANGE.

Name: ADSR

Full Name: Administration of Designed Services Review tag

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD.CKT

Description: This tag indicates that the work for a circuit requires Designed Services review.

Name: CDD

Full Name: Change of Due Date

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD

Description: This tag indicates that this contract represents only a Change of Due Date.

Name: CKID

Full Name: Circuit Identifier

Type: Tag

Size: Maximum of 60 characters

Data Type: Alphanumeric plus "-", ";", ".", and ",."

Appears in: FOMSRR.*FOMSRR.STDRQ.HDATA

Description: This tag specifies the ID of a facility to identify a particular circuit.

Name: CKT

Full Name: Circuit

Type: Aggregate

Appears in: FOMSTC.*ORD.CKT

Description: The CKT aggregate contains all connection and disconnection instructions for one circuit.

Name: CLS

Full Name: Class of Service

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: R | B | C

Appears in: FOMSTC.*ORD.CKT

Description: R= Residential Service, B=Business Service, C=Coin Service.

Name: CP

Full Name: Cable Pair

Type: Tag

Size: Maximum of 15 characters

Data Type: Alphanumeric plus "-"

Appears in: FOMSRR.*FOMSRR.STDRQ.HDATA

Description: This tag specifies the ID of a facility to identify a particular circuit.

Name: CTTYPE

Full Name: Circuit ID Type

Type: Tag

Size: Maximum of 5 characters

Data Type: Alphanumeric

Code Set: TN | CKID | OE | CP

Appears in: FOMSTC.*ORD.CKT, FOMSTC.*ORD.CKT.RELORD

Description: This tag specifies the type of facility whose ID will follow.

Name: CTID

Full Name: Circuit Identifier

Type: Tag

Size: Maximum of 60 characters

Data Type: Alphanumeric plus "-", ";", ".", and ",."

Appears in: FOMSTC.*ORD.CKT, FOMSTC.*ORD.CKT.RELORD

Description: This tag specifies the ID of a facility which identifies a particular circuit.

Name: DD

Full Name: Due Date

Type: Tag

Size: 8 characters

Data Type: Numeric

Code Set: yyyymmdd

Appears in: FOMSTC.*ORD

Description: This tag specifies the due date of the order.

Name: DEBUG

Full Name: Debug data

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRS

Description: This aggregate will contain formatted output from FOMS when the debug flag has been set.

Name: FAC

Full Name: Facility

Type: Aggregate

Appears in: FOMSTC.*ORD.CKT.IN/OUT/REU

Description: This contains the type and ID of a facility to be connected, disconnected or reused.

Name: FDD

Full Name: Frame Due Date

Type: Tag

Size: 8 characters

Data Type: Numeric

Code Set: yyyymmdd

Appears in: FOMSTC.*ORD

Description: This tag specifies the date by which the “framework” (connections or disconnections) must be done.

Name: FDT

Full Name: Frame Due Time

Type: Tag

Size: 1-5 characters

Data Type: Alphanumeric

Appears in: FOMSTC.*ORD

Description: This tag specifies the time by which the “framework” (connections or disconnections) must be done.

Name: FOMSDBG

Full Name: FOMS Debug Flag

Type: Tag

Size: 3 characters

Data Type: Alphanumeric

Code Set: 000-FFF

Appears in: FOMSRR.*FOMSRR.PLHDR

Description: This tag is the hexadecimal representation that is used to turn on the debug feature in FOMS. The setting determines how detailed the debug output will be. The recommended setting is FFF.

Name: *FOMSRR

Full Name: *FOMS Request Response

Type: Section

Appears in: FOMSRR contract

Description: This section will be used in both the request and response contracts between FOMS and a REMS. Some aggregates will appear when the contract is a request from a REMS to FOMS, and others will appear when the contract is a response from FOMS to a REMS.

Name: FOMSRSP

Full Name: FOMS Response Flag

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: N

Appears in: FOMSRR.*FOMSRR.PLHDR

Description: This tag must be set to 'N' if the sending system does not want a response contract from FOMS when the request contract has executed successfully.

Name: FRLOC

Full Name: Frame location

Type: Tag

Size: 5-7 characters

Data Type: Alphanumeric

Code Set: FXX-XXX

Appears in: FOMSTC.*ORD.IN/OUT/REU

Description: This tag specifies the three character working frame ID and the 1-3 character zone where the facility is terminated, separated by a dash.

Name: HDATA

Full Name: H-line Data

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRQ

Description: This aggregate contains h-line data for the specified transaction.

Name: ID

Full Name: Facility Identifier

Type: Tag

Size: Maximum of 60 characters

Data Type: Alphanumeric plus "-", ";", ".", and ",".

Appears in: FOMSTC.*ORD.CKT.IN/OUT/REU.FAC

Description: This tag specifies the ID of a facility to be connected, disconnected or reused.

Name: IDATA

Full Name: I-line Data

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRQ

Description: If present, this aggregate contains all the I-line data required to run the requested FOMS transaction.

Name: IN

Full Name: In Aggregate

Type: Aggregate

Appears in: FOMSTC.*ORD.CKT

Description: The IN aggregate contains connection instructions for two facilities.

Name: JR

Full Name: Jeopardy Reason Code

Type: Tag

Size: Maximum of 3 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.STDRQ.HDATA

Description: This tag specifies the jeopardy reason code to be placed on a circuit.

Name: LB

Full Name: Loopback

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD.CKT.IN, FOMSTC.*ORD.CKT.OUT,
FOMSTC.*ORD.CKT.REU

Description: This tag indicates that a connection or disconnection is loopback.

Name: LCLID

Full Name: Local ID

Type: Tag

Size: 1-250 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.PLHDR

Description: This tag may be used to match requests from the source system to responses from the destination system.

Name: LINE

Full Name: Line of output from a FOMS transaction

Type: Tag

Size: 1-72 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.STDERR, FOMSRR.*FOMSRR.STDOUT,
FOMSRR.*FOMSRR.DEBUG

Description: This tag represents one line of output from the requested FOMS transaction.

Name: LST

Full Name: Line and Station Transfer

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD.CKT

Description: This tag indicates that this circuit represents an embedded line and station transfer.

Name: MP

Full Name: Multipoint

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD.CKT.IN, FOMSTC.*ORD.CKT.OUT,
FOMSTC.*ORD.CKT.REU

Description: This tag indicates that a connection or disconnection is multipoint.

Name: NOSCM

Full Name: No SCM Indicator

Type: Tag

Size: 3 characters

Data Type: Alphabetic

Code Set: MOD | CAN | CDD | MDD

Appears in: FOMSTC.*ORD.CKT

Description: This tag indicates that the contract contains work for a circuit that has been modified (MOD), cancelled (CAN), had a due date or frame due date change (CDD), or had both a modification and a DD/FDD change (MDD) before it was frame completed.

Name: NPANXX

Full Name: FOMS Wire Center Identification

Type: Tag

Size: 6 characters

Data Type: Numeric

Appears in: FOMSRR.*FOMSRR.STDRS

Description: This tag represents the primary NPANXX of the wire center and is used as the wire center ID.

Name: NUML

Full Name: Number of Lines

Type: Tag

Size: 1-3 characters

Data Type: Numeric

Appears in: FOMSRR.*FOMSRR.STDERR, FOMSRR.*FOMSRR.STDOUT,
FOMSRR.*FOMSRR.DEBUG

Description: This tag represents the number of LINE tags that follow.

Name: OC

Full Name: Order Category

Type: Tag

Size: 1-4 characters

Data Type: Alphabetic

Code Set: BTH, CMB, COR, CPC, HOT, LNP, LS, LLNF, UNE

Appears in: FOMSTC.*ORD.CKT

Description: This tag specified the order category for a circuit.

Name: OCTTYPE

Full Name: Old Circuit ID Type

Type: Tag

Size: Maximum of 5 characters

Data Type: Alphanumeric

Code Set: TN | CKID | OE | CP

Appears in: FOMSTC.*ORD.CKT

Description: This tag specifies the type of facility whose ID will follow.

Name: OCTID

Full Name: OLD Circuit Identifier

Type: Tag

Size: Maximum of 60 characters

Data Type: Alphanumeric plus "-", ";", ".", and ",."

Appears in: FOMSTC.*ORD.CKT

Description: This tag specifies the ID of the facility which previously identified a particular circuit.

Name: ODATA

Full Name: O-line Data

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRQ

Description: If present, this aggregate contains all the O-line data required to run the requested FOMS transaction.

Name: OE

Full Name: Office or Line Equipment

Type: Tag

Size: Maximum of 20 characters

Data Type: Alphanumeric plus "-"

Appears in: FOMSRR.*FOMSRR.STDRQ.HDATA

Description: This tag specifies the ID of a facility to identify a particular circuit.

Name: OP

Full Name: Option

Type: Tag

Size: 3 characters

Data Type: Alphanumeric

Code Set: RJR

Appears in: FOMSRR.*FOMSRR.STDRQ.HDATA

Description: This tag specifies that the order/circuit is to be removed from jeopardy.

Name: ORD

Full Name: Order Section

Type: Section

Appears in: FOMSTC contract

Description: This section contains all connection and disconnection instructions for one order.

Name: ORDNO

Full Name: Order Number

Type: Tag

Size: Maximum of 20 characters

Data Type: Alphanumeric

Appears in: FOMSTC.*ORD, FOMSTC.*ORD.CKT.RELORD

Description: This tag specifies the ID of the order for which connection data is being sent.

Name: OT

Full Name: Order Type

Type: Tag

Size: 2-3 characters

Data Type: Alphabetic

Code Set: CD|CH|F|NC|T|CIO|LST|MCT|SET|WAO

Appears in: FOMSTC.*ORD, FOMSTC.*ORD.CKT.RELORD

Description: This tag specifies the type of order for which connection data is being sent, or the type of a related order.

Name: OUT

Full Name: Out Aggregate

Type: Aggregate

Appears in: FOMSTC.*ORD.CKT

Description: The OUT aggregate contains disconnection instructions for two facilities

Name: PLHDR

Full Name: Platform Header

Type: Aggregate

Appears in: FOMSTC.*ORD

Description: This aggregate contains local administrative data which is passed from the source system to the destination system and returned.

Name: PRI

Full Name: Priority

Type: Tag

Size: 1 character

Data Type: Numeric

Code Set: 0-3

Appears in: FOMSTC.*ORD

Description: The priority tag indicates the level of urgency of making the connections/disconnections in this contract. PRI=0 indicates an MCT; PRI=1 indicates a cancellation or modification to a service order which has been sent to the REMS previously, or a service order due today; PRI=2 is for all other service orders and modifications and cancellations of work orders; PRI=3 is for work orders.

Name: RDATA

Full Name: Remark Line Data

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRQ

Description: If present, this aggregate contains all the R-line data required to run the requested FOMS transaction.

Name: REDIRECT

Full Name: Redirect Output Flag

Type: Tag

Size: 1-14 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.PLHDR

Description: This tag indicates that the debug output should be redirected to a separate file and, optionally, name the file. If present, it will be set to a 'Y' or a 1-14 character file name. If it is set to 'Y', FOMS will name the file. If REDIRECT is not present, the debug output will be interspersed with the standard output in the STDOUT aggregate.

Name: RELORD

Full Name: Related Order

Type: Aggregate

Appears in: FOMSRR.*ORD.CKT

Description: If present, this aggregate contains the order number and circuit ID of a dependent related order or circuit, which has not been completed by the REMS.

Name: REU

Full Name: Reuse aggregate

Type: Aggregate

Appears in: FOMSTC.*ORD.CKT

Description: This aggregate contains the IDs of two facilities that are connected together and should remain so.

Name: RMK

Full Name: Remark

Type: Tag

Size: 1-33 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.STDRQ.RDATA

Description: This tag represents a remark to be placed on the circuit in jeopardy.

Name: SPECFUNC

Full Name: Special Functionality

Type: Tag

Size: 1-4 characters

Data Type: Alphanumeric

Appears in: FOMSTC.*ORD.CKT.IN/OUT/REU.FAC

Description: This tag represents the type of functionality that applies to a particular facility in an assembly.

Name: SSM

Full Name: Special Safeguarding Measures

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD.CKT

Description: This tag indicates that special safeguarding measures are required for a circuit.

Name: SSP

Full Name: Special Service Protection

Type: Tag

Size: 1 character

Data Type: Alphabetic

Code Set: Y

Appears in: FOMSTC.*ORD.CKT

Description: This tag indicates that special service protection is required for a circuit.

Name: STATUS

Full Name: Status Identifier

Type: Tag

Size: 2 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.STDRS

Description: •This tag contains the status of the request; 0=totally successful, nonzero = unsuccessful or partially successful.

- 0=successful, no warning, no debug output; only STDOUT returned.
 - 1=partially successful, with at least one error/warning; STDOUT and STDERR returned.
 - 2=error, no debug output; only STDERR returned.
 - 3=successful with debug output; STDOUT and DEBUG returned.
 - 4=partially successful with at least one error/warning and debug output; STDOUT, STDERR and DEBUG returned.
 - 5=error with debug output; STDERR and DEBUG returned.
 - 6=debug output only; only DEBUG returned.
-

Name: STDERR

Full Name: Standard Error

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRS

Description: This aggregate contains an error message from FOMS.

Name: STDOUT

Full Name: Standard Output

Type: Aggregate

Appears in: FOMSRR.*FOMSRR.STDRS

Description: This aggregate contains all the output for the requested FOMS transaction.

Name: STDRQ

Full Name: Standard Request

Type: Aggregate

Appears in: FOMSRR.*FOMSRR

Description: This aggregate contains all the data necessary to execute a specified transaction in FOMS.

Name: STDRS

Full Name: Standard Response

Type: Aggregate

Appears in: FOMSRR.*FOMSRR

Description: This aggregate contains all the output (standard output and error messages from the requested FOMS transaction.

Name: TN

Full Name: Telephone Number

Type: Tag

Size: Maximum of 12 characters

Data Type: Numeric plus ”-”

Appears in: FOMSRR.*FOMSRR.STDRQ.HDATA

Description: This tag specifies the ID of a facility to identify a particular circuit.

Name: TRAN

Full Name: Transaction Name

Type: Tag

Size: 3 characters

Data Type: Alphanumeric

Appears in: FOMSRR.*FOMSRR.STDRQ, FOMSRR.*FOMSRR.STDRS

Description: This tag specifies the name of the transaction to be executed in FOMS.

Name: TYPE

Full Name: Type of Facility

Type: Tag

Size: Maximum of 5 characters

Data Type: Alphanumeric

Code Set: BL, CCPT, CP, ICE, ME, OE, TKP, TP, TRE

Appears in: FOMSTC.*ORD.CKT.IN/OUT/REU.FAC

Description: This tag specifies the type of facility to be connected, disconnected or reused.

Name: WC

Full Name: Wire Center Name

Type: Tag

Size: 2-11 characters

Data Type: Alphanumeric

Appears in: FOMSTC.*ORD

Description: This tag specifies the external wire center designation.
