

INTERNATIONAL SEMATECH



Recipe and Parameter Management (RaP) Evaluation Method

International SEMATECH Manufacturing Initiative
Technology Transfer #08094959A-TR

Advanced Materials Research Center, AMRC, International SEMATECH Manufacturing Initiative, and ISMI are servicemarks of SEMATECH, Inc. **SEMATECH** and the **SEMATECH** logo are registered servicemarks of SEMATECH, Inc. All other servicemarks and trademarks are the property of their respective owners.

Recipe and Parameter Management (RaP) Evaluation Method
Technology Transfer #08094959A-TR
International SEMATECH Manufacturing Initiative
September 25, 2008

Abstract: This document from the MFGM054M project provides a basic evaluation method for systems using Recipe and Parameter Management (RaP) capabilities as defined in SEMI E139. The evaluation method focuses on normal operations only. This evaluation method can be performed with the International SEMATECH Manufacturing Initiative (ISMI) RaP Reference Implementation (RRI) software package.

Keywords: Automated Data Collection, Equipment Performance, Factory Automation, Recipe Management, Standards Conformance

Authors: Lance Rist (ISMI), Konrad Rosenbaum (PEER Group), and Thomas Marsh (PEER Group)

Approvals: Lance Rist, Author
Harvey Wohlwend, Project Manager
Olaf Rothe, Program Manager
Joe Draina, Director
Laurie Modrey, Technology Transfer Team Leader

Table of Contents

1	INTRODUCTION	1
2	PURPOSE.....	2
3	SCOPE.....	2
4	RELATED RESOURCES	2
5	ABBREVIATIONS, TERMINOLOGY, AND METRICS DEFINITIONS	3
	5.1 Abbreviations	3
	5.2 Terminology.....	3
6	DESCRIPTION OF THE RAP EVALUATION METHOD.....	4
	6.1 Overview	4
	6.2 RaP Reference Implementation (Rri).....	4
7	EVALUATION PREPARATION	4
	7.1 Assessment and Evaluation Resources	5
	7.2 Evaluation Scope and Parameters	6
	7.3 Recipe Configuration	8
8	EVALUATION EXECUTION	9
	8.1 General Instructions	9
	8.1.1 Execution Steps.....	10
	8.1.2 RRI Usage Tips.....	11
	8.2 Evaluating Equipmentnodes	11
	8.2.1 Part A – Setup	11
	8.2.2 Part B – Tracking of Local Changes.....	11
	8.2.3 Part C – Service Functionality Evaluation.....	11
	8.2.4 Part D – Job Execution.....	12
	8.2.5 Parts E and F – Evaluation Subject as Client.....	12
	8.3 Evaluation Steps to be Performed at Ficsnodes	12
	8.3.1 Part A – Setup	13
	8.3.2 Part B – Service Functionality Evaluation.....	13
	8.3.3 Parts C and D – Evaluation Subject as Client.....	13
	8.4 Evaluation Steps to be Performed at Pde Editornodes.....	13
	8.4.1 Part A – Setup	13
	8.4.2 Part B – Service Functionality Evaluation.....	14
	8.4.3 Parts C and D – Evaluation Subject as Client.....	14
9	ANALYSIS AND REPORTING	14
	9.1 Data Analysis	15
	9.2 Reporting.....	15
10	SUMMARY.....	17
	APPENDIX A – EVALUATION PROCEDURES: EQUIPMENTNODES.....	18
	APPENDIX B – EVALUATION PROCEDURES: FICSNODES	32
	APPENDIX C – EVALUATION PROCEDURES: EDITORNODES.....	41

List of Tables

Table 1	Recipe Store Change Events and Parameters.....	7
Table 2	Traceability Events and Parameters	7
Table 3	Recipes to be Available on Equipment for RaP Evaluation	9
Table A-1	EquipmentNode Evaluation Procedures.....	19
Table B-1	FICSnode Evaluation Procedures.....	33
Table C-1	EditorNode Evaluation Procedures	42

1 INTRODUCTION

SEMI E139, *Specification for Recipe and Parameter Management (RaP)*, delineates the interaction between the Factory Information and Control System (FICS) and the equipment to manage the specifications (recipes) of equipment processing. RaP supports the following goals:

- Trusted recipe content, a guarantee that the recipe on the equipment is exactly the one the factory approved, downloaded, and/or selected
- Enhanced process control through flexible user-defined parameter settings, and clear visibility of parameters and their defaults in the recipe
- Reduced number of recipes due to enhanced parameterization opportunities and multi-part recipe component reuse
- Traceability of recipe and parameter usage through event reporting supported by guaranteed unique recipe identifiers
- Detailed and efficient verification process to ensure recipe validity
- Support of advanced factory-side recipe management approaches
- Improved support for off-tool recipe editors, paving the way for universal recipe editors, through a common interface definition using mainstream communication technology

With the goal of encouraging the uniform adoption and implementation of RaP, this document defines a common evaluation method. It provides procedures to evaluate the RaP capabilities of equipment, editors, and FICS systems using both SECS and SOAP/XML-based implementations of SEMI E139. The evaluation benefits implementers by reducing development effort and complexity, resulting in lower cost and higher quality.

References provide the background to understand the foundations of this method. The primary source is SEMI E139.

RaP consists of a primary SEMI standard and its related implementation sub-specifications as listed below. The SEMI E139 specification is available from SEMI. It includes the sub-specifications.

- SEMI E139 – *Specification for Recipe and Parameter Management (RaP)*: Defines a standard means for an equipment supplier to create a model of their equipment's recipes.
- SEMI E139.1 – *XML Schema for the RaP PDE*: Builds on E139 and describes the exact encoding of RaP compliant recipe headers and containers.
- SEMI E139.2 – *SECS-II Protocol for Recipe and Parameter Management*: Defines the communication protocol for the use of SECS-II (E5) for RaP.
- SEMI E139.3 – *SOAP Binding for Recipe and Parameter Management*: Defines an alternative communication protocol for RaP based on SEMI E132 and the Interface A approach using XML/SOAP-based messaging.

This document relies upon the ISMI approach to communicating with a RaP-enabled equipment as presented in the ISMI *RaP Usage Scenarios* (Technology Transfer #08054929A-TR). See Section 4 for a list of related documents.

2 PURPOSE

The ISMI RaP Evaluation Method is a guideline for a common approach to early and effective determination of basic equipment capabilities and performance against the SEMI RaP standards and end-user expectations. The mission of the RaP Evaluation Method is to facilitate consistent evaluations of the RaP interface and to demonstrate the ability to support minimum scenarios. It should accelerate the delivery of high quality functionality to end-users.

ISMI developed this method to evaluate implementations of the SEMI RaP standards. The method is *not* an in-depth conformance test against the RaP standards; such a test is expected to become available in the future. Instead, it is meant to ensure that implementers have correctly developed their implementations and to detect significant divergence in interpretation of the standards at an early point in the commercial lifecycle. Evaluation results should be treated as confidential and used in a constructive improvement process, not as a conformance metric.

3 SCOPE

The ISMI RaP Evaluation Method is intended to confirm a limited set of scenarios and possible variations for each of standard RaP functions. This set should represent the expected “typical” use by end-users. The evaluation method is not intended to cover all possible situations in RaP communication such as error-handling situations, which still requires more extensive feedback from the field.

Successful completion of the ISMI RaP Evaluation Method does not signify compliance with the RaP standards. The main output of the evaluation will be a report that lists the issues detected during the evaluation. There is no overall “pass/fail” result.

The ISMI RaP Evaluation Method does not include a process for improving the supplier’s RaP implementation and achieving standard conformance. However, it would be a useful part of such a process.

4 RELATED RESOURCES

Several related documents are available www.ismi.sematech.org, while others (such as industry standards) can be found at www.semi.org. The following documents are directly referenced in this document:

1. *Equipment Client Connection Emulator (ECCE)*, software application available for download from <http://www.cimetrix.com>
2. *Recipe and Parameter Management (RaP) Usage Scenarios*, ISMI, Technology Transfer #08054929A-TR, <http://ismi.sematech.org/docubase/abstracts/4929atr.htm>
3. *RaP Reference Implementation (RRI)*, software application, available for download from <http://ismi.sematech.org/rri>.
4. SEMI E30, *Generic Model for Communications and Control of Manufacturing Equipment (GEM)*
5. SEMI E132, *Specification for Equipment Client Authentication and Authorization*
6. SEMI E134, *Specification for Data Collection Management*

7. SEMI E139, *Specification for Recipe and Parameter Management (RaP)*
8. SEMI E139.1, *XML Schema for the RaP PDE*
9. SEMI E139.2, *SECS-II Protocol for Recipe and Parameter Management*
10. SEMI E139.3, *SOAP Binding for Recipe and Parameter Management*

5 ABBREVIATIONS, TERMINOLOGY, AND METRICS DEFINITIONS

5.1 Abbreviations

ACL	– Access Control List
EDA	– Equipment Data Acquisition (synonym for “Interface A”)
FICS	– Factory Information and Control System (ITRS term)
PDE	– Process Definition Element
RaP	– Recipe and Parameter Management
RMS	– Recipe Management System
RRI	– RaP Reference Implementation (See Section 4)
UID	– Unique Identifier

5.2 Terminology

The following terms are defined as they are used within this document. In many cases, a more general term is defined in the context of the RaP evaluation method. Terms from the SEMI E139 RaP standard are too numerous to be included in this section. The reader should reference that document as needed.

Evaluation Subject	– Implementation of RaP on which the evaluation method is to be run. This may be an equipment, a recipe editor, or a factory system application.
Evaluation Tool	– Software application used to execute the RaP Evaluation Method.
Interface A	– Refers to the equipment data acquisition (EDA) interface defined by SEMI E120, E125, E128, E132, and E134.
Process Definition Element (PDE)	– The smallest unit of a recipe that can be handled independently.
RaP client	– An entity that is able to use the services provided by RaP. A single entity may be both a RaPnode and a RaP client.
RaPnode	– An entity/program that provides RaP services to RaP clients.
Recipe	– The collection of all PDEs that are needed to execute a process at the equipment. This can be a single executable PDE or a collection of PDEs.
TransferContainer	– Compressed collection of PDEs for transfer to/from a RaP-enabled application (Equipment, FICS, Recipe Editor).
Usage Scenario	– A description of how the services defined in the RaP standards may be used to accomplish a specific task. A Usage Scenario typically consists of a sequence of messages between a client and a server under a specified set of conditions.
Variation	– Application of a usage scenario or a portion of a usage scenario under different sets of conditions that lead to different outcomes. Variations typically represent small changes to the condition sets.

6 DESCRIPTION OF THE RAP EVALUATION METHOD

6.1 Overview

The ISMI RaP evaluation process is comprised of three phases:

- 1) **Preparation** – these steps that should be performed before arrival at the location the evaluation will be performed (see Section 7).
- 2) **Execution** – the on-site execution of the evaluation is the primary activity of performing the evaluation procedure steps (see Section 8).
- 3) **Reporting** – upon completion of the evaluation, the evaluation results are summarized (see Section 9).

The following are some of the target outcomes of a RaP evaluation:

- Confirmation that the RaP client can successfully connect to the target RaPnodes using a SECS or XML/SOAP implementation
- Evaluation of the RaPnodes' ability to perform the basic RaP functions
- Understanding of the details of the RaP implementation, including recipe structures, use of RaP messaging, and user interaction with the system

6.2 RaP Reference Implementation (Rri)

This evaluation method does not specify which software tool should be used to perform the evaluation procedures. Any software tool that meets the requirements in Section 7.1 can be used.

The RRI is the recommended utility for executing the RaP Evaluation Method. Section 4 provides a link to download the RRI. Included with the evaluation procedures is additional information specific to using the RRI.

When the document refers to the “evaluation tool,” it means any software used to execute the evaluation method. Directions and tips relating to the RRI are included.

7 EVALUATION PREPARATION

To perform a RaP evaluation effectively, certain logistical and infrastructure requirements must be met. Because the evaluation method may require travel to the evaluation site, this section defines steps that should be taken ahead of time to ensure a smooth on-site evaluation.

The subsections below are formulated as checklists to be completed before moving to the evaluation site.

7.1 Assessment and Evaluation Resources

This method was created with the assumption that the following resources are available:

- Evaluation technician/team - technical personnel knowledgeable about RaP standards, SECS communication, and, as appropriate, the XML/SOAP protocol.
- Subject of the evaluation - a functional semiconductor tool, editor, or host-side application with a RaP implementation.
 - Time for the evaluation must be reserved for the evaluation subject. In particular, where production equipment is evaluated, advance arrangements must be made.
 - A carrier with three wafers should be available to be used in the processing run of the equipment.
 - In some cases, additional resources are needed. This must be determined and arrangements made.
- Evaluation tool - a software application installed on a computer (usually a laptop) that will be used as the communication partner of the evaluation subject. Note that the RRI is a satisfactory evaluation tool. The evaluation tool must satisfy the following requirements:
 - Maintains a time-stamped log of all messages sent and received on a per-session basis (including all data contained in the messages)
 - Allows the user to create/store/send instances of standard messages containing specified data in the message attributes
 - Is able to verify and generate the necessary UIDs/GIDs and checksums required by the RaP standard
 - Is able to generate, transmit, and evaluate the TransportContainers specified by the RaP standard
 - Is able to send and receive all RaP service requests and able to handle the responses
 - Supports the RaP and data collection protocols needed for the evaluation (see Section 7.2)
- Event reporting configuration tool – Event reporting configuration cannot be performed with the RRI tool. If Interface A is to be used for event reporting, the ECCE tool can be used. For SECS, multiple commercial tools are available. The supplier of the evaluation target will typically have an adequate tool available.
- Documentation – Acquire and review equipment automation documentation and any release notes relating to the specific model/version of the evaluation subject.
 - This document will typically be the source for the event and variable identifiers specified in Section 7.2
- Evaluation subject expertise – Arrange for the participation of appropriate supplier personnel based on the expertise level of the evaluation technician.

- Previous evaluation method execution performed by supplier – Perform the evaluation method steps independently in advance of the on-site evaluation. In this way, a high degree of success for the evaluation can be assured.

7.2 Evaluation Scope and Parameters

The scope of the evaluation must be determined beforehand and information gathered to support that scope.

- RaP protocols to be evaluated – Note that if both protocols are chosen, the expectation is that the evaluation will be run twice, once for each protocol.
 - RaP SECS protocol support (per SEMI E139.2)
 - RaP XML/SOAP protocol support (per SEMI E139.3)
- Event data collection method – When the evaluation subject is an equipment, determine which mechanism will be used for event reporting of RaP traceability events and recipe store change events.
 - SECS/GEM event reporting (SEMI E30)
 - Interface A event reporting (SEMI E134)
- Communication connections – Where multiple protocols are used, determine whether multiple physical communications connections are required. This might indicate the need for additional network hardware (e.g., a switch) or cabling.
- Determine the connectivity information for the protocols to be used:
 - For SECS protocol support, the following are pertinent for RaPnodes:
 - IP address
 - TCP port
 - Device ID
 - Timeouts, according to the requirements of the evaluation environment
 - For Interface A, the following connection details are pertinent:
 - IP address
 - URL for necessary services (typically, E139 RaP and E132 Session Management)
- Event identification – Where the evaluation subject is an equipment, determine that the needed events and data are available for the evaluation.

Fill in the values for the events and data items in Table 1 and Table 2. Table 1 lists the recipe store change events and the data items to be reported with each of those events. Table 2 lists the traceability events and the data items to be reported for each of those events. Traceability events/data are defined in SEMI E139 R3 (Related Information).

In these two tables, fill in only the column for the event-reporting mechanism to be used (SECS or Interface A).

Table 1 Recipe Store Change Events and Parameters

Event/Parameter	SECS	Interface A
Recipe Store Change Events		
PDEadded event	<i>CEID:</i>	<i>sourceId:</i> <i>eventId:</i>
PDEremoved event	<i>CEID:</i>	<i>sourceId:</i> <i>eventId:</i>
PDElist	<i>DVVAL:</i>	<i>sourceId:</i> <i>parameterId:</i>
RecipeStoreChangeTime	<i>SVID:</i>	<i>sourceId:</i> <i>parameterId:</i>
ResolvePDEreferences	<i>ECID:</i>	<i>sourceId:</i> <i>parameterId:</i>

Table 2 Traceability Events and Parameters

Event/Parameter	SECS	Interface A
Traceability Events	<input type="checkbox"/> <i>Execution Events support</i> <input type="checkbox"/> <i>PDE parameter Changed events supported</i>	
PDE Execution Events (one per recipe type)	<i>CEID:</i>	<i>sourceId:</i> <i>eventId:</i>
PDE Parameter Change Events (one per recipe type)¹	<i>CEID:</i>	<i>sourceId:</i> <i>eventId:</i>
RecipeComponentID	<i>DVVAL:</i>	<i>sourceId:</i> <i>parameterId:</i>
SubstratesAffected	<i>DVVAL:</i>	<i>sourceId:</i> <i>parameterId:</i>
EqpComponentID	<i>DVVAL:</i>	<i>sourceId:</i> <i>parameterId:</i>
ParametersApplied	<i>DVVAL:</i>	<i>sourceId:</i> <i>parameterId:</i>
PJID (Process Job ID)	<i>DVVAL:</i>	<i>sourceId:</i> <i>parameterId:</i>

¹ Note that this is a relatively new feature and is not included in the body of the Evaluation Method. It is included here to document whether it currently exists on the equipment.

Although not generally available at this time, evaluation technicians should also investigate the possibility of including PDEparameter changed events (see SEMI E139 Related Information). Like the traceability events described above, one event may be available for each PDE type on the equipment. However, these are not formal constituents of the evaluation method at this time. The evaluation technician should determine whether monitoring of these events should be included as part of the actual evaluation.

7.3 Recipe Configuration

The evaluation method refers specifically to named recipes defined in Table 3. To support all the evaluation steps, the hierarchical, multi-part recipes should be used where supported by the equipment. Additionally, the recipes should form a representative set of the different recipe types available on the equipment. Since the types of recipe used for each equipment can vary significantly, the selection must depend on the evaluation technician's judgment.

Note that since the evaluation method calls for processing scenarios to be executed, one recipe (REM-Exec) should be of short duration and should not perform destructive processes (i.e., should perform only "dry-run" actions).

A set of recipes must be prepared and available on the equipment before the evaluation. Each must be represented by a master PDE (with the executable attribute set to true). If multi-part recipes are supported by the equipment, then the recipe structure of each sample recipe should be representative of the different recipe types available on the equipment. Unless otherwise specified, RaP messages that reference PDEs should do so by *gid* rather than *uid*. The evaluation technician must include a description of the recipe structures in the report, including the name of each sub-PDE used by these master PDEs.

It is left to the judgment of the evaluation technician if certain of the evaluation procedures should be performed for each of the recipe types on the equipment.

If the equipment supports only single-part recipes, some RaP messages, such as resolvePDE() might seem not to apply. This is not the case. Since device makers prefer to execute the same scenarios on all equipment if possible, all equipment should be capable of processing all messages. Therefore, all parts of the evaluation method should be executed in all cases (except where specifically noted in the evaluation definitions).

The recipes should be set up and available on the equipment before the evaluation. To facilitate the evaluation method, recipes are identified by standardized names (see Table 3). As part of the evaluation, the evaluation technician must record the actual names of these PDEs and for multi-part recipes, their hierarchical structure.

In addition, several PDEs must be prepared to be used by the evaluation tool, but must not be present on the equipment at the start of the evaluation. These PDEs must be named "REM-Send1" to "REM-SendN" (where the recipes are named "REM-Send1," "REM-Send2," up to "REM-SendN"). These PDEs will be used in the evaluation as part of the sendPDE request to send multiple PDEs to the equipment from the evaluation tool.

Table 3 Recipes to be Available on Equipment for RaP Evaluation

Master PDE Name	Notes	PDE Hierarchy (To be filled in by evaluator.)
REM-Test	The number of child PDEs should be determined according to typical use patterns on the equipment to be evaluated. If it is a multi-part recipe, at least one PDE reference should be by GroupID to a Group that contains REM-Test1 and REM-Test2, with REM-Test2 being a newer revision of REM-Test1.	
REM-Delete	Note that this recipe will be deleted during the evaluation method.	
REM-Rec1 to REM-RecN	Represents multiple unrelated recipes (named "REM-Rec1," "REM-Rec2," etc.) for use in multiple-recipe transfers. At least three unrelated recipes should be provided according to this nomenclature.	
REM-Exec	Short duration dry-run recipe for job execution scenarios. Should be designed to be executed with ResolvePDEreferences set to TRUE.	

During the evaluation, the evaluation technician is asked to create PDEs on the equipment during part B of the evaluation of EquipmentNodes (see Table A-1, Step EQP-B1). This PDE, to be named REM-Local1, can have any content that is acceptable to the evaluation subject. In a subsequent step, REM-Local2 is to be created, based on REM-Local1, as a new member of the same group (same gid). REM-Local2 will have exactly the same content as REM-Local1, except for the name field.

8 EVALUATION EXECUTION

The RaP evaluation is performed to characterize the interface through a systematic process. The subsequent sections describe the set of activities that need to be performed to execute the evaluation method. These consist of general instructions applicable to all RaPnodes followed by instructions specific to each RaPnode. Tables of detailed evaluation steps are in the appendices. This section describes the intent of each grouping of steps and provides additional information or instructions that are beyond the scope of the tables.

8.1 General Instructions

The evaluation method consists of a series of evaluation steps documented in the appendices. One appendix is provided for each RaPnode, as follows:

- Appendix A – Evaluation Procedures: Equipmentnodes
- Appendix B – Evaluation Procedures: FICSnodes
- Appendix C – Evaluation Procedures: EditorNodes

The three appendices contain equivalent procedures that have been tailored to the specific needs and capabilities for each type of RaPnode. For example, because the FICSnode does not support the deletePDE() method, the FICSnode procedure does not include steps for evaluating deletePDE().

The appendices are structured as a linear set of steps, which must be run sequentially. Each step represents a feature to be evaluated, a procedure to perform the evaluation, and an expectation result to be compared to the actual result. There are three major types of evaluation steps:

- 1) Preparation
- 2) Service evaluation
- 3) Evaluation subject as client

8.1.1 Execution Steps

During the evaluation, each step should be performed as described in the tables and the results or observations should be noted in the space provided. Three possible outcomes of an evaluation step are as follows:

- The step may be performed with the expected results.
- The step may be performed, but with unexpected results.
- The step may be skipped because it is unsupported by the target RaPnode or explicitly excluded business reasons.

The tables have space record notes relating to the evaluation step results. Other observations should also be recorded for the final report. During the evaluation, the evaluation technician must retain a record of the following:

- Notes pertaining to preparation of the environment, including all necessary changes to settings of the evaluation tool and evaluation subject.
- The specific recipes used during the evaluation and any modifications needed outside those specified in the evaluation method.
- Log files from the evaluation tool.
- Log files from the evaluation subject, if available.
- Any observations during the evaluation that relate to the operations performed, including exceptional behavior, unexpected warnings, or errors, etc.

This information, together with the checklists from Section 7, and the completed table from the appropriate appendix must be included in the final evaluation report (see Section 9).

If more than one RaPnode is being evaluated, they should be evaluated separately, each receiving its own evaluation method execution and report.

When performing multi-protocol evaluations, the evaluation procedures should be executed for SECS-based and Interface A-based communication separately by performing the complete sequence of steps on the RaPnode for each protocol. Although mixed use of the two protocols is a reasonable implementation approach, this evaluation method does not evaluate such use.

During a RaP evaluation, significant factors outside the evaluation technician's control could materially affect the evaluation process. If such factors are encountered, the evaluation technician will need to determine whether or not to continue the evaluation. If a decision is made to terminate the evaluation, all factors contributing to the decision should be identified, captured, and reported in the final evaluation report. At that point, the evaluation technician focuses on documenting the achievements and failures of the activities completed to that point.

8.1.2 RRI Usage Tips

Although it is not required, this evaluation method is best performed with the RRI. The user is expected to be familiar with usage of the RRI. The tables within the appendices include detailed usage tips specific to the RRI indicating the location where actions can be performed with the RRI. The RRI usage tips in some cases also include special notes and warnings peculiar to the RRI implementation.

8.2 Evaluating Equipmentnodes

This section describes activities specific to performing the evaluation for RaP EquipmentNodes. The detailed procedural steps are in Appendix A. The following sections are organized according to the evaluation step groupings.

The evaluation tool should function as an FICSnode for this part of the evaluation.

8.2.1 Part A – Setup

The initial step must establish communications and verify the most basic levels of functionality necessary for further evaluations. This step establishes whether the equipment is ready for the evaluation, establishes communication to the equipment, and performs initial configuration.

8.2.2 Part B – Tracking of Local Changes

This group of evaluation steps involves making direct changes to the local equipment recipe store (e.g., creating, modifying, and deleting recipes locally on the equipment). During this procedure, it is expected that the necessary PDEadded and PDERemoved events arrive and contain the correct UIDs and timestamps. This is intended to establish whether the host system is informed about actions taken locally by an operator or engineer.

8.2.3 Part C – Service Functionality Evaluation

This step evaluates whether all required RaP services on the equipment work as expected:

- getPDEDirectory
 - Report of the directory structure and attributes, with and without filtering of PDEs is performed. Combinations of filters are checked for returning the correct subsets.
- getPDE
 - Retrieval of single PDEs and groups of PDEs is performed.
- getPDEheader
 - The retrieval of header information of PDEs is performed.

- deletePDE
 - The remote deletion of PDEs and some failure modes of this service are checked.
- sendPDE
 - Transfer of a PDE to the equipment is performed. It is also evaluated whether the RecipeStoreChangeTime is set correctly by the equipment.
- resolvePDE
 - These steps evaluate the behavior of the resolvePDE service, which is used for the following purposes:
 - Determining which PDE is used for which reference/GID when the host sets up jobs with GIDs as RecID.
 - Determining the complete hierarchy of what PDEs are referenced by a Master PDE.
 - Determining whether a PDEmap contains all necessary references for a job setup to succeed if ResolvePDEreference is set to FALSE (as recommended by ISMI).
- verifyPDE
 - The multiple variations of the verifyPDE service are executed to check the validity of PDEs.
 - The service can check a single PDE or a hierarchy of PDEs. Even if the equipment only supports single-part recipes, it is still necessary to run all steps to ensure that the equipment does not reject verification requests with depth = all.

8.2.4 Part D – Job Execution

The job execution procedures are intended to determine whether the traceability events are produced and can provide the specified data. The recipe traceability is described in Related Information 3 of SEMI E139. As described in Section 7, it is highly recommended that a recipe be used during these steps that is of a short duration. Wafers (see Section 7.1) will be required at this stage.

8.2.5 Parts E and F – Evaluation Subject as Client

The procedures in Parts E and F verify that the evaluation subject correctly implements and interoperates with the evaluation tool when it initiates service requests. For example, this functionality can normally be initiated at the equipment user interface to upload or download recipes. The RaP services invocation may not be direct and will vary among equipment implementations. Performing these evaluation steps may require assistance from the supplier. Setup and service functionality evaluation are defined for this case.

8.3 Evaluation Steps to be Performed at Ficsnodes

The evaluation tool should function as an EquipmentNode for this part of the evaluation.

8.3.1 Part A – Setup

The initial step must establish communications and verify the most basic levels of functionality necessary for further evaluations. This step establishes whether the FICSnode is ready for the evaluation, establishes communication between the equipment and the FICS, and performs initial configuration.

8.3.2 Part B – Service Functionality Evaluation

This step evaluates whether all required RaP services on the FICS work as expected:

- getPDEDirectory
 - The directory structure and attributes, with and without filtering PDEs, are reported. Combinations of filters are checked for returning the correct subsets.
- getPDE
 - Single PDEs and groups of PDEs are retrieved.
- getPDEheader
 - The header information of PDEs is retrieved.
- sendPDE
 - A PDE is transferred to the equipment. Whether the storage timer is set correctly by the equipment is also evaluated.

8.3.3 Parts C and D – Evaluation Subject as Client

The procedures in Parts C and D verify that the evaluation subject correctly implements and interoperates with the evaluation tool when it initiates service requests. This functionality should be possible to initiate from the FICS. The RaP services invocation may not be direct and will vary among FICS implementations. The performance of these evaluation steps may require assistance from the supplier.

Setup and service functionality evaluation are defined for this case.

8.4 Evaluation Steps to be Performed at Pde Editornodes

The evaluation tool should function as an EquipmentNode for this part of the evaluation.

8.4.1 Part A – Setup

The initial step must establish communications and verify the most basic levels of functionality necessary for further evaluations. This step establishes whether the EditorNode is ready for the evaluation, establishes communication between the equipment and the editor, and performs initial configuration.

8.4.2 Part B – Service Functionality Evaluation

This step evaluates whether all required RaP services on the FICS work as expected:

- getPDEDirectory
 - The directory structure and attributes, with and without filtering PDEs, are reported. Selected combinations of filters are checked.
- getPDE
 - Single PDEs and groups of PDEs are retrieved.
- getPDEheader
 - The header information of PDEs is retrieved.
- deletePDE
 - The remote deletion of PDEs and some failure modes of this service are checked.
- sendPDE
 - A PDE is transferred to the equipment. Whether the storage timer is set correctly by the equipment is also evaluated.

8.4.3 Parts C and D – Evaluation Subject as Client

The procedures in Parts C and D verify that the evaluation subject correctly implements and interoperates with the evaluation tool when it initiates service requests. For example, this functionality can normally be initiated at the editor user interface to upload or download recipes. The RaP services invocation may not be direct and will vary among editor implementations. Performing these evaluation steps may require assistance from the supplier.

Setup and service functionality evaluation are defined for this case.

9 ANALYSIS AND REPORTING

The RaP evaluation report should capture the results of the evaluation based on the procedures, checklists, and forms in this document.

The evaluation should provide a consistent method for analyzing and reporting results. Some examples are as follows:

- a list of features evaluated
- a list of evaluated features that meet the expectations of a specific evaluation
- a list of features not evaluated
- a list of features that did not meet evaluation expectations
- comparisons to industry requirements

9.1 Data Analysis

Once the evaluation has been completed, the data must be analyzed. This analysis must be thorough and measured against expectations about each item. The analysis must determine whether the goals and expectations established by the method were met.

Most analysis is based on the communication log of the RaP client session. However, certain data must be recorded during the evaluation. Procedures² will be provided to enable data captured during an evaluation to be analyzed. During the evaluation, the approximate time of each major portion of the evaluation procedure should be recorded to simplify the later job of analyzing the log. Evaluations should include an examination of the equipment's log files to verify expected functionality and identify any errors or anomalies encountered by the equipment.

Note that the evaluation technician must not only verify that messages have been successfully sent, but also inspect their content for validity. This is especially true for PDE transfers. The evaluation technician must inspect the content of the PDE headers and the manifest of the TransferContainer that have been transferred during the exchange of getPDE and sendPDE messages.

9.2 Reporting

Any evaluation that characterizes or compares performance results to end-user expectations must be formally documented. This section outlines the content, structure, and documentation of equipment evaluation results. It also provides a high-level outline of a final report summarizing the evaluation activities.

The final evaluation report is an overall summary of the evaluation results. Supplier comments and inputs on the results or improvement plans and roadmaps can be included in the final evaluation report. The final report will reference detailed data gathered during the process; raw data should be provided to the supplier upon request.

The purpose of the equipment evaluation is to produce valid objective results that are representative of the equipment at its current stage of development and documented in the final report. The final evaluation report is a comprehensive summary of all the evaluation results from activities and information resulting from the evaluation. The report provides a comprehensive review of the equipment's RaP capabilities, including a minimum of detailed data. The following information should be included in the final report:

- Equipment model number, serial number(s), options, configuration, and software release number or identifier.
- Evaluation results for each of the phases of the evaluation.
- A listing of any problems encountered and an analysis of its relationship to RaP standards requirements.

² In this version of the RaP evaluation method, the procedures for data analysis have not yet been defined.

An outline of the content and structure to be used for an evaluation report is as follows:

- **Executive Summary**
 - Brief overview of the evaluation results with enough context for understanding
- **Introduction**
 - What was the evaluation subject (include version)
 - What evaluation tool was used (include version)
 - When and where was the evaluation done
- **Evaluation Preparation**
 - For each item described in the Section 7 checklists and instructions, describe what was done and what decision was made. For example, was the “previous evaluation method execution” performed and what were the results; were all the needed events and parameters identified before the evaluation; etc.
- **Evaluation Results**
 - Include the information recorded on the procedure pages from the appendices in the order and form provided.
 - Merge any more detailed comments that pertain to specific parts or steps of the process that were recorded separately.
- **Evaluation Analysis**
 - Based on the evaluation results, discuss those steps that did not perform as expected and provide supporting information from the log files or other sources. If possible, identify the root cause of each case of non-performance. Assign a severity to each based on the importance of the feature that is affected, the degree to which the feature is affected, and the availability of a workaround.
- **Issues Summary**
 - Summarize the root problems defined above to be addressed by the provider of the evaluation subject.
- **Supplier Inputs**
 - Preliminary results should be made available to the supplier. The supplier should be invited to comment in writing on unexpected results, outline plans for updates or other next steps, and otherwise respond to the evaluation in general. The supplier’s inputs should be included unedited as long as they are of reasonable length and do not include marketing material.
 - This section should be clearly marked as coming from the supplier and presented in an unedited form.
- **Conclusion**
 - Discuss whether the evaluation subject is adequate to the anticipated need and what, if any, plan for improvement exists. Outline any anticipated next steps.

- **Appendix**
 - Include the evaluation tool log files and, if available, the logs from the evaluation subject. Also include any other raw data available as appropriate, including notes taken during the evaluation. It is also helpful to include copies of the recipe files and example TransferContainer files if possible.

The evaluation report should be treated as a confidential document. The intent is not to publicize supplier issues, but rather to give constructive feedback and encourage the supplier to recognize and address the issues.

10 SUMMARY

The evaluation methodology contained herein includes end-user expectations and methods necessary to evaluate a supplier's RaP implementation. The ISMI RaP Evaluation Method is a process to efficiently determine equipment capabilities against the expectations of the SEMI E139 standard. The mission of a RaP evaluation is to enable accelerated and timely delivery of high quality functionality to end-users. The value of a consistent, standardized approach that meets the needs of the industry and reduces the cost of redundant evaluation resources cannot be overstated.

Appendix A – Evaluation Procedures: Equipmentnodes

The following table details the procedure to execute the RaP Evaluation Method for EquipmentNodes. Note that rows highlighted in blue are intended to be performed on the equipment, not from the evaluation tool. Please ensure that the prerequisite activities in Section 7 have been performed and that Section 8 has been read and understood before beginning the evaluation.

The following columns are defined in the table:

- **Category** – high level description of the procedure step(s)
- **ID** – a unique identifier for the given step
- **Evaluation Procedure** – actions to perform to execute the procedure step
- **Expected Result** – details mechanism to verify procedure completion; a check box is provided for the use of the evaluation technician
- **RRI Tips** – usage tips specific to the RRI to accomplish the evaluation procedure
- **Comments** – any notes or observations regarding the execution or results of the evaluation procedure.

The appendix is designed to be printed and used to record evaluation results. However, supplemental observations must also be recorded outside of this table for use in final reporting.

Evaluation Date: _____

Evaluation Technician: _____

Equipment Supplier (make/model): _____

Comments:

Table A-1 EquipmentNode Evaluation Procedures

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
A. Setup						
Preparation	EQP-A1	Configure evaluation tool to receive PDEadded and PDEremoved events with the necessary variables attached (See Section 7).	<input type="checkbox"/>	None.	Run the RRI FICS Simulator. Configuration settings found in menu Options → Configuration . From the configuration dialog window, navigate to the SECS Configuration tab. Enter the appropriate host and port information. Note: Upon configuring the RRI, it is recommended to save your configuration for future reuse.	
	EQP-A2	Ensure the PDE storage area of the client is empty.	<input type="checkbox"/>	None.	Create a new storage area by updating configuration settings found in menu Options → Configuration . From the configuration dialog window, navigate to the PDE Repository tab. It is highly recommended to start a new log session, which is performed from the Session Log tab of the main window.	
Connect	EQP-A3	Establish communication and/or session with the equipment. The evaluation tool must run as an FICSnode.	<input type="checkbox"/>	Check connections used: <input type="checkbox"/> SECS/HSMS (S1F13 successful) <input type="checkbox"/> RaP-XML/SOAP <input type="checkbox"/> Interface A-XML/SOAP	The simulator is started by the Start Simulator button located on the status bar at the bottom of the main window.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
Configuration	EQP-A4	Check that the necessary PDEs are on the equipment using the getPDEdirectory() message with the name attribute. (See Section 7.3). Note: A PDE named "REM-Local1" should not be present on the equipment, as it will be created later, during EQP-B1.	<input type="checkbox"/>	All necessary PDEs are available	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EQP-A5	Check that the equipment is in the correct state to be used for the remainder of the evaluation.	<input type="checkbox"/>	Using SECS, the tool is in an online / remote state. With Interface A, it should be verified that the user has the necessary privileges to perform data collection. The equipment is in a production-ready state (e.g., not be in a maintenance mode) and is idle.	This cannot be performed with the RRI, but can be verified from the tool user interface.	
	EQP-A6	Set up appropriate data collection reports for the configured events and variables (see Section 7.2, Table 1, and Table 2. With SECS protocol, enable these events. With Interface A, define and activate a DCP containing these events.	<input type="checkbox"/> <input type="checkbox"/>	Data collection reports are defined on the equipment. PDEparameter Changed events supported and enabled.	Note that this action cannot be performed with the RRI, and should be accomplished with an external tool (see Section 7.1).	
B. Tracking Local Changes						
Create	EQP-B1	Create and save a PDE locally on the equipment. Call this PDE "REM-Local1". The content of this PDE is not important. (see Section 7.3).	<input type="checkbox"/>	PDEadded event received by the evaluation tool with the <i>uid</i> of the new PDE	Navigate to the Equipment (Remote) tab and find the Recipe Store Change events section on the SECS Status sub-tab.	
	EQP-B2	From the evaluation tool, send the getPDEdirectory() message requesting the name attribute to the equipment.	<input type="checkbox"/>	getPDEdirectory() response result contains "REM-Local1"	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
Modify	EQP-B3	Modify PDE "REM-Local1" locally on the equipment by executing the following steps: <ul style="list-style-type: none"> - Copy the PDE - Change the PDE name to "REM-Local2" - Save "REM-Local2" with the original as an antecedent and with the same <i>gid</i>, but different <i>uid</i>. 	<input type="checkbox"/>	PDEadded event received with the <i>uid</i> of the new PDE version associated	Navigate to the Equipment (Remote) tab and find the Recipe Store Change events section on the SECS Status sub-tab.	
	EQP-B4	From the evaluation tool, request the PDE directory by sending the getPDEdirectory() message to the equipment requesting the name attribute.	<input type="checkbox"/>	getPDEdirectory() result contains "REM-Local1" and "REM-Local2" PDEs.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
Delete	EQP-B5	Delete "REM-Local1" and "REM-Local2" locally on the equipment.	<input type="checkbox"/>	PDEremoved event received with the <i>uid</i> of the deleted PDE associated	Navigate to the Equipment (Remote) tab and find the Recipe Store Change events section on the SECS Status sub-tab.	
	EQP-B6	Verify deletion by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the equipment.	<input type="checkbox"/>	getPDEdirectory() result no longer contains the "REM-Local1" or "REM-Local2" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
Manual Copy <i>Perform only if the equipment allows PDEs to be transferred by removable media.</i>	EQP-B7	Copy PDE "REM-Delete" from the equipment onto an external medium (e.g., a network drive or USB memory stick). Note: record the PDEs original location.	<input type="checkbox"/>	None.		
	EQP-B8	Delete file(s) representing the "REM-Delete" PDE locally on the equipment.	<input type="checkbox"/>	PDEremoved event received by the evaluation tool with the <i>uid</i> of the "REM-Delete" PDE when it is deleted.	Navigate to the Equipment (Remote) tab and find the Recipe Store Change events section on the SECS Status sub-tab.	
	EQP-B9	Copy the PDE back to its original location on the equipment.	<input type="checkbox"/>	PDEadded event received by the evaluation tool with the <i>uid</i> of the "REM-Delete" PDE when it is copied back.	Navigate to the Equipment (Remote) tab and find the Recipe Store Change events section on the SECS Status sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
C. Service Functionality Evaluation						
getPDEdirectory	EQP-C1	Request the directory with all attributes and no filters by sending the getPDEdirectory() message from the evaluation tool to the equipment. Make note of the author of the "REM-Test" PDE for use in <i>EQP-C5</i> .	<input type="checkbox"/>	getPDEdirectory() response provides full directory including all attributes.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EQP-C2	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the equipment: <ul style="list-style-type: none"> - Attribute = "UID" - Filter type = "EQ" - Value = <uid of "REM-Test"> 	<input type="checkbox"/>	getPDEdirectory() response list includes only the "REM-Test" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EQP-C3	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the equipment: <ul style="list-style-type: none"> - Attribute = "name" - Filter type = "Like" - Value = "REM-" 	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which include the string "REM-" in their name, including the REM-Test PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EQP-C4	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name , author , and createDate attributes from the evaluation tool to the equipment: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <author from REM-Test PDE> <p>Include the author attribute in the getPDEdirectory() request to validate the result.</p>	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which match the filter, including the REM-Test PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab. Note: Be aware that dates need to be typed manually; the format is an XML Schema <i>dateTime</i> formatted string.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-C5	Request the PDE directory using the following combination of filters by sending the getPDEdirectory() message with the author and gid attributes from the evaluation tool to the equipment. 1st Filter: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <string not equal to author of REM-Test> 2nd Filter: <ul style="list-style-type: none"> - Attribute = "GID" - Filter type = "EQ" - Value = <GID of "REM-Test"> 	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which match the filter and does not include the REM-Test PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab. Note: Be aware that dates need to be typed manually; the format is an XML Schema <i>dateTime</i> formatted string.	
getPDE	EQP-C6	From the evaluation tool, request the "REM-Get1" PDE from the equipment using the getPDE() message.	<input type="checkbox"/>	The equipment returns the "REM-Get1" PDE in the getPDE() response, and the evaluation tool stores the PDE locally.	From the RaP Requests tab, navigate to the Get PDE sub-tab. Note: In RRI you must first get the PDE directory before the remote PDE list will be populated in the Get PDE tab.	
	EQP-C7	From the evaluation tool, request the PDEs "REM-Get1" to "REM-GetN" (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDE() message requesting the multiple PDEs.	<input type="checkbox"/>	The equipment returns PDEs "REM-Get1" to "REM-GetN" in the getPDE() response, and the evaluation tool stores the PDEs locally.	From the RaP Requests tab, view the Response area of the Get PDE sub-tab.	
getPDEheader	EQP-C8	From the evaluation tool, request the header of the "REM-Get1" PDE from the equipment using the getPDEheader() message.	<input type="checkbox"/>	The equipment returns the "REM-Get1" PDE header in the getPDEheader() response.	From the RaP Requests tab, navigate to the Get PDE Header sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-C9	From the evaluation tool, request the headers of the PDEs named "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDEheader() message request.	<input type="checkbox"/>	The equipment returns the headers for PDEs named "REM-Get1" to "REM-GetN" in the getPDEheader() response.	From the RaP Requests tab, view the Response area of the Get PDE Header sub-tab.	
deletePDE	EQP-C10	Upload the PDE named "REM-Delete" to the evaluation tool by sending the getPDE() message from the evaluation tool to the equipment.	<input type="checkbox"/>	The equipment returns the "REM-Delete" PDE in the getPDE() response.	From the RaP Requests tab, navigate to the Get PDE sub-tab.	
	EQP-C11	Delete PDE named "REM-Delete" from the equipment by sending the deletePDE() message from the evaluation tool to the equipment.	<input type="checkbox"/>	<input type="checkbox"/> The deletePDE() successfully executes with no errors. <input type="checkbox"/> PDEremoved event received by the evaluation tool with the uid of "REM-Delete"	From the RaP Requests tab, navigate to the Delete PDE sub-tab.	
	EQP-C12	Verify that the "REM-Delete" PDE is no longer resident on the equipment by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the equipment.	<input type="checkbox"/>	The getPDEdirectory() response shows that "REM-Delete" is not among PDEs on the equipment.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
sendPDE	EQP-C13	For the "REM-Delete" PDE, send the requestToSendPDE() from the evaluation tool to the equipment.	<input type="checkbox"/>	The requestToSendPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Request To Send PDE sub-tab.	
	EQP-C14	For the "REM-Delete" PDE, send the sendPDE() from the evaluation tool to the equipment.	<input type="checkbox"/>	The sendPDE() message executes with no errors. PDEadded event is received by the evaluation tool.	From the RaP Requests tab, navigate to the Send PDE sub-tab.	
	EQP-C15	Send the getPDEdirectory() message from the evaluation tool to the equipment and verify that the "REM-Delete" PDE arrived at the tool. Include the name attribute in the getPDEdirectory() request.	<input type="checkbox"/>	The getPDEdirectory() response includes the "REM-Delete" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EQP-C16	For the "REM-Delete" PDE, send the requestToSendPDE() and sendPDE() from the evaluation tool to the equipment. Note that this is an intentional repeat of <i>EQP-C13 & EQP-C14</i> .	<input type="checkbox"/>	The sendPDE() message executes with no errors. PDEadded event is received by the evaluation tool (new copy replaces the old copy).	From the RaP Requests tab, look at the Response area of the Send PDE sub-tab	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-C17	Send the requestToSendPDE() message from the evaluation tool to the equipment with the PDEs "REM-Send1" to "REM-SendN" as an argument (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3).	<input type="checkbox"/>	The requestToSendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Request to Send PDE sub-tab	
	EQP-C18	Send the sendPDE() message from the evaluation tool to the equipment with the PDEs "REM-Send1" to "REM-SendN" as an argument.	<input type="checkbox"/>	The sendPDE() message executes with no errors. PDEadded event is received by the evaluation tool.	From the RaP Requests tab, look at the Response area of the Send PDE sub-tab	
	EQP-C19	Verify the existence of PDEs "REM-Send1" to "REM-SendN" by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the equipment.	<input type="checkbox"/>	The getPDEdirectory() response includes the PDEs "REM-Send1" to "REM-SendN".	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EQP-C20	Delete PDEs "REM-Send1" to "REM-SendN" by sending the deletePDE() message from the evaluation tool to the equipment referencing these PDEs.	<input type="checkbox"/>	The deletePDE() successfully executes with no errors. PDEremoved event is received by the evaluation tool.	From the RaP Requests tab, look at the Response area of the Delete PDE sub-tab.	
resolvePDE	EQP-C21	Set the ResolvePDEreferences equipment constant to the value TRUE.	<input type="checkbox"/>	None.	Navigate to the Equipment (Remote) tab and then to the SECS Status sub-tab. Fill in the Resolve PDE References checkbox and press the Write button to set the constant on the equipment.	
	EQP-C22	Send the resolvePDE() request from the evaluation tool to the equipment with the <i>uid</i> of the "REM-Test" PDE (a master PDE, where the Executable attribute is set to TRUE) as the targetPDE parameter without an inputMap .	<input type="checkbox"/>	The resolvePDE() response outputMap produces a valid structure for "REM-Test" which includes "REM-Test1". Note: If the equipment supports only single-part recipes, the outputMap will contain only REM-Test.	Navigate to the RaP Requests tab and choose the Resolve PDE sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-C23	Send resolvePDE() request from the evaluation tool to the equipment with the <i>uid</i> of the "REM-Test2" PDE. Note: This evaluation step should be skipped if the equipment supports only single-part recipes.	<input type="checkbox"/>	The resolvePDE() response outputMap produces the appropriate structure for child "REM-Test2".	Navigate to the RaP Requests tab and choose the Resolve PDE sub-tab.	
	EQP-C24	Set the ResolvePDEreferences equipment constant to the value FALSE.	<input type="checkbox"/>	None.	Navigate to the Equipment (Remote) tab and then to the SECS Status sub-tab. Uncheck the Resolve PDE References checkbox and press the Write button to set the constant on the equipment.	
	EQP-C25	Send the resolvePDE() request from the evaluation tool to the equipment with the <i>uid</i> of the "REM-Test" PDE (a master PDE, where the Executable attribute is set to TRUE) as the targetPDE parameter. Include the "REM-Test" structure in the inputMap as a parameter. The inputMap must include an entry for the group id shared by "REM-Test1" and "REM-Test2," and which resolves to "REM-Test1".	<input type="checkbox"/>	The resolvePDE() response outputMap produces a valid structure for "REM-Test" which includes "REM-Test1". Note: If the equipment supports only single-part recipes, the outputMap will contain only REM-Test.	To configure an inputMap , navigate to the Input Map tab. To send the resolvePDE request, navigate to the RaP Requests tab and choose the Resolve PDE sub-tab.	
verifyPDE	EQP-C26	From the evaluation tool send a verifyPDE() message to the equipment for the "REM-Test" master PDE with the following parameters: <ul style="list-style-type: none"> - Depth = all - Type = checksum - inputMap = <same as for EQP-C25> 	<input type="checkbox"/>	The verifyPDE() message executes with no errors and includes checksum in its response.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step. Note: In RRI you must first get the PDE directory before the remote PDE list will be populated in the Get PDE tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-C27	From the evaluation tool send a verifyPDE() message to the equipment for the "REM-Test" master PDE with the following parameters: <ul style="list-style-type: none"> - Depth = all - Type = validate - inputMap = <same as for EQP-C25> 	<input type="checkbox"/>	The verifyPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step.	
	EQP-C28	From the evaluation tool send a verifyPDE() message to the equipment for the "REM-Test" master PDE with the following parameters: <ul style="list-style-type: none"> - Depth = single - Type = checksum - inputMap = <same as for EQP-C25> 	<input type="checkbox"/>	The verifyPDE() message executes with no errors and includes checksum in its response.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step.	
	EQP-C29	From the evaluation tool send a verifyPDE() message to the equipment for the "REM-Test" master PDE with the following parameters: <ul style="list-style-type: none"> - Depth = single - Type = validate - inputMap = <same as for EQP-C25> 	<input type="checkbox"/>	The verifyPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step.	
	EQP-C30	Set the ResolvePDEreferences equipment constant to the value TRUE.	<input type="checkbox"/>	None.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step.	
	EQP-C31	From the evaluation tool send a verifyPDE() message to the equipment for the "REM-Test2" PDE with the following parameters: <ul style="list-style-type: none"> - Depth = single - Type = validate 	<input type="checkbox"/>	The verifyPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-C32	From the evaluation tool send a verifyPDE() message to the equipment for the "REM-Test2" PDE with the following parameters: <ul style="list-style-type: none"> - Depth = all - Type = checksum 	<input type="checkbox"/>	The verifyPDE() message executes with no errors and includes the checksum in its response.	From the RaP Requests tab, navigate to the Verify PDE sub-tab to perform the verification step.	
D. Job Execution						
Setup	EQP-D1	Set up a process job locally to execute "REM-Exec". The carrier with three wafers will be used in this step (see Section 7.1)	<input type="checkbox"/>	The job should be set up successfully.		
Run	EQP-D2	Start the job.	<input type="checkbox"/> <input type="checkbox"/>	The job should run to completion. Confirm the traceability events in the log for each type of recipe for which events were configured. (see Section 7.2)		
E. Evaluation Subject as Client - Setup						
Preparation	EQP-E1	Configure equipment to connect to the evaluation tool and ensure the evaluation tool is running as an FICSnode.	<input type="checkbox"/>	None.	Run the RRI FICS Simulator. Configuration settings found in menu Options → Configuration . From the dialogue window, navigate to the SECS Configuration tab. At this time, also enter the appropriate host and port information. Note: Upon configuring the RRI, it is recommended to save your configuration for future reuse.	
Connect	EQP-E2	From the equipment, establish communication and/or session with the evaluation tool.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Check connections used: SECS/HSMS (S1F13 successful) RaP-XML/SOAP Interface A-XML/SOAP	The simulator is first started by the Start Simulator button located on the status bar at the bottom of the main window.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
Configuration	EQP-E3	Check that the necessary PDEs are on the FICS using the getPDEdirectory() message with the name attribute from the evaluation tool. (See Section 7.3)	<input type="checkbox"/>	All necessary PDEs are available		
F. Evaluation Subject as Client - Service Functionality Evaluation						
getPDEdirectory	EQP-F1	Request the directory with all attributes and no filters by sending the getPDEdirectory() message from the equipment to the evaluation tool. Make note of the author of the "REM-Test" PDE for use in <i>EQP-C5</i> .	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	EQP-F2	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the equipment to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "UID" - Filter type = "EQ" - Value = <uid of "REM-Test"> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	EQP-F3	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the equipment to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "name" - Filter type = "Like" - Value = "REM-" 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	EQP-F4	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name and createDate attributes from the equipment to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <author from REM-Test PDE> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EQP-F5	Request the PDE directory using the following combination of filters by sending the getPDEdirectory() message with the author and gid attributes from the equipment to the evaluation tool. 1st Filter: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <string not equal to author of REM-Test> 2nd Filter: <ul style="list-style-type: none"> - Attribute = "GID" - Filter type = "EQ" - Value = <GID of "REM-Test"> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
getPDE	EQP-F6	From the equipment, request the "REM-Get1" PDE from the evaluation tool using the getPDE() message.	<input type="checkbox"/>	The correctly formed getPDE() message is received, prompting the evaluation tool to return the "REM-Get1" PDE to the equipment. PDEadded event is received by the evaluation tool.		
	EQP-F7	From the equipment, request the PDEs "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDE() message requesting the multiple PDEs.	<input type="checkbox"/>	The correctly formed getPDE() message is received, prompting the evaluation tool to return the "REM-Get1" to "REM-GetN" PDEs to the equipment. PDEadded event is received by the evaluation tool.		
getPDEheader	EQP-F8	From the equipment, request the header of the "REM-Get1" PDE from the evaluation tool using the getPDEheader() message.	<input type="checkbox"/>	The correctly formed getPDEheader() message is received, prompting the evaluation tool to return the "REM-Get1" header to the equipment.		
	EQP-F9	From the evaluation tool, request the headers of the PDEs named "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDEheader() message request.	<input type="checkbox"/>	The correctly formed getPDEheader() message is received, prompting the evaluation tool to return the headers for the "REM-Get1" to "REM-GetN" PDEs to the equipment.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
sendPDE	EQP-F10	For the "REM-Delete" PDE, send the requestToSendPDE() from the equipment to the evaluation tool.	<input type="checkbox"/>	The correctly formed requestToSendPDE() message is received at the evaluation tool.		
	EQP-F11	For the "REM-Delete" PDE, send the sendPDE() from the equipment to the evaluation tool.	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDE is transferred.		
	EQP-F12	Send the getPDEdirectory() message from the equipment to the evaluation tool and verify that the "REM-Delete" PDE arrived at the evaluation tool. Include the name attribute in the getPDEdirectory() request.	<input type="checkbox"/>	The getPDEdirectory() is received, and the response includes the "REM-Delete" PDE. Note, if this is not possible to verify using the getPDEdirectory() message, it may still be possible to inspect manually on the evaluation tool.		
	EQP-F13	For the "REM-Delete" PDE, send the requestToSendPDE() and sendPDE() from the equipment to the evaluation tool. Note that this is an intentional repeat of <i>EQP-C13</i> and <i>EQP-C14</i> .	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDE is transferred.		
	EQP-F14	Send the requestToSendPDE() message from the equipment to the evaluation tool with the PDEs "REM-Send1" to "REM-SendN" as an argument (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3).	<input type="checkbox"/>	The correctly formed requestToSendPDE() message is received at the evaluation tool.		
	EQP-F15	Send the sendPDE() message from the equipment to the evaluation tool with the PDEs "REM-Send1" to "REM-SendN" as an argument.	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDEs are transferred.		
	EQP-F16	Verify the existence of PDEs "REM-Send1" to "REM-SendN" by sending the getPDEdirectory() message requesting the name attribute from the equipment to the evaluation tool.	<input type="checkbox"/>	The getPDEdirectory() is received, and the response includes the "REM-Send1" to "REM-SendN" PDEs. Note, if this is not possible to verify using the getPDEdirectory() message, it may still be possible to inspect manually on the evaluation tool.		

Appendix B – Evaluation Procedures: FICSnodes

The following table details the procedure to execute the RaP Evaluation Method for FICSnodes. Note that rows highlighted in blue are intended to be performed on the FICS, not from the evaluation tool. Please ensure that the prerequisite activities in Section 7 have been performed and that Section 8 has been read and understood before beginning the evaluation.

The following columns are defined in the table:

Category – high level description of the procedure step(s)

ID – a unique identifier for the given step

Evaluation Procedure – actions to perform to execute the procedure step

Expected Result – details mechanism to verify procedure completion; a check box is provided for the use of the evaluation technician

RRI Tips – usage tips specific to the RRI to accomplish the evaluation procedure

Comments – any notes or observations regarding the execution or results of the evaluation procedure.

The appendix is designed to be printed and used to record evaluation results. However, supplemental observations must also be recorded outside of this table for use in final reporting.

Evaluation Date: _____

Evaluation Technician: _____

Equipment Supplier (make/model): _____

Comments:

Table B-1 FICSnode Evaluation Procedures

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
A. Setup						
Preparation	FICS-A1	Ensure the PDE storage area of the client is empty.	<input type="checkbox"/>	None.	Create a new storage area by updating configuration settings found in menu Options → Configuration . From the configuration dialog window, navigate to the PDE Repository tab. It is highly recommended to start a new log session, which is performed from the Session Log tab of the main window.	
Connect	FICS-A2	Establish communication and/or session with the FICS. The evaluation tool must run as an EquipmentNode.	<input type="checkbox"/>	Check connections used: <input type="checkbox"/> SECS/HSMS (S1F13 successful) <input type="checkbox"/> RaP-XML/SOAP <input type="checkbox"/> Interface A-XML/SOAP	The simulator is started by the Start Simulator button located on the status bar at the bottom of the main window.	
Configuration	FICS-A3	Check that the necessary PDEs are on the FICS using the getPDEdirectory() message with the name attribute from the evaluation tool. (See Section 7.3)	<input type="checkbox"/>	All necessary PDEs are available	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
B. Service Functionality Evaluation						
getPDEdirectory	FICS-B1	Request the directory with all attributes and no filters by sending the getPDEdirectory() message from the evaluation tool to the FICS. Make note of the author of the "REM-Test" PDE for use in <i>FICS-C5</i> .	<input type="checkbox"/>	getPDEdirectory() response provides full directory including all attributes. The FICS should return all PDEs for the equipment selected for this evaluation, it should not return PDEs for other equipment in the factory.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	FICS-B2	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the FICS: <ul style="list-style-type: none"> - Attribute = "UID" - Filter type = "EQ" - Value = <uid of "REM-Test"> 	<input type="checkbox"/>	getPDEdirectory() response list includes only the "REM-Test" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	FICS-B3	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the FICS: <ul style="list-style-type: none"> - Attribute = "name" - Filter type = "Like" - Value = "Recipe" 	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which include the string "Recipe" in their name, including the REM-Test PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	FICS-B4	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name, author and createDate attributes from the evaluation tool to the FICS: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <author from REM-Test PDE> <p>Include the author attribute in the getPDEdirectory() request to validate the result.</p>	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which match the filter, including the REM-Test PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab. Note: Be aware that dates need to be typed manually; the format is an XML Schema <i>dateTime</i> formatted string.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	FICS-B5	<p>Request the PDE directory using the following combination of filters by sending the getPDEdirectory() message with the author and gid attributes from the evaluation tool to the FICS.</p> <p>1st Filter:</p> <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <string not equal to author of REM-Test> <p>2nd Filter:</p> <ul style="list-style-type: none"> - Attribute = "GID" - Filter type = "EQ" - Value = <GID of "REM-Test"> 	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which match the filter and does not include the REM-Test PDE.	<p>Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.</p> <p>Note: Be aware that dates need to be typed manually; the format is an XML Schema <i>dateTime</i> formatted string.</p>	
getPDE	FICS-B6	From the evaluation tool, request the "REM-Get1" PDE from the FICS using the getPDE() message.	<input type="checkbox"/>	The FICS returns the "REM-Get1" PDE in the getPDE() response, and the evaluation tool stores the PDE locally.	<p>From the RaP Requests tab, navigate to the Get PDE sub-tab.</p> <p>Note: In RRI you must first get the PDE directory before the remote PDE list will be populated in the Get PDE tab.</p>	
	FICS-B7	From the evaluation tool, request the PDEs "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDE() message requesting the multiple PDEs.	<input type="checkbox"/>	The FICS returns PDEs "REM-Get1" to "REM-GetN" in the getPDE() response, and the evaluation tool stores the PDEs locally.	From the RaP Requests tab, view the Response area of the Get PDE sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
getPDEheader	FICS-B8	From the evaluation tool, request the header of the "REM-Get1" PDE from the FICS using the getPDEheader() message.	<input type="checkbox"/>	The FICS returns the "REM-Get1" PDE header in the getPDEheader() response.	From the RaP Requests tab, navigate to the Get PDE Header sub-tab.	
	FICS-B9	From the evaluation tool, request the headers of the PDEs named "REM-Get1" to "REM-GetN" (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDEheader() message request.	<input type="checkbox"/>	The FICS returns the headers for PDEs named "REM-Get1" to "REM-GetN" in the getPDEheader() response.	From the RaP Requests tab, view the Response area of the Get PDE Header sub-tab.	
sendPDE	FICS-B10	For the "REM-Delete" PDE, send the requestToSendPDE() from the evaluation tool to the FICS.	<input type="checkbox"/>	The requestToSendPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Request To Send PDE sub-tab.	
	FICS-B11	For the "REM-Delete" PDE, send the sendPDE() from the evaluation tool to the FICS.	<input type="checkbox"/>	The sendPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Send PDE sub-tab.	
	FICS-B12	Send the getPDEdirectory() message from the evaluation tool to the FICS and verify that the "REM-Delete" PDE arrived at the FICS. Include the name attribute in the getPDEdirectory() request.	<input type="checkbox"/>	The getPDEdirectory() response includes the "REM-Delete" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	FICS-B13	For the "REM-Delete" PDE, send the requestToSendPDE() and sendPDE() from the evaluation tool to the FICS. Note that this is an intentional repeat of <i>FICS-B10 and FICS-B11</i> .	<input type="checkbox"/>	The sendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Send PDE sub-tab	
	FICS-B14	Send the requestToSendPDE() message from the evaluation tool to the FICS with the PDEs "REM-Send1" to "REM-SendN" as an argument (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3).	<input type="checkbox"/>	The requestToSendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Request to Send PDE sub-tab	
	FICS-B15	Send the sendPDE() message from the evaluation tool to the FICS with the PDEs "REM-Send1" to "REM-SendN" as an argument.	<input type="checkbox"/>	The sendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Send PDE sub-tab	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	FICS-B16	Verify the existence of PDEs "REM-Send1" to "REM-SendN" by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the FICS.	<input type="checkbox"/>	The getPDEdirectory() response includes the PDEs "REM-Send1" to "REM-SendN".	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	FICS-B17	Delete PDEs "REM-Send1" to "REM-SendN" by sending the deletePDE() message from the evaluation tool to the FICS referencing these PDEs.	<input type="checkbox"/>	The deletePDE() successfully executes with no errors.	From the RaP Requests tab, look at the Response area of the Delete PDE sub-tab.	
C. Evaluation Subject as Client – Setup						
Preparation	FICS-C1	If not using the pre-existing connection from the previous evaluation cases, configure FICS to connect to the evaluation tool and ensure the evaluation tool is running as an EquipmentNode.	<input type="checkbox"/>	None.	Run the RRI FICS Simulator. Configuration settings found in menu Options → Configuration . From the dialogue window, navigate to the SECS Configuration tab. At this time, also enter the appropriate host and port information. Note: Upon configuring the RRI, it is recommended to save your configuration for future reuse.	
Connect	FICS-C2	If not already connected, from the FICS establish communication and/or session with the evaluation tool.	<input type="checkbox"/>	Check connections used: <input type="checkbox"/> SECS/HSMS (S1F13 successful) <input type="checkbox"/> RaP-XML/SOAP <input type="checkbox"/> Interface A-XML/SOAP	The simulator is first started by the Start Simulator button located on the status bar at the bottom of the main window.	
Configuration	FICS-C3	Check that the necessary PDEs are on the equipment using the getPDEdirectory() message with the name attribute from the evaluation tool. (See Section 7.3)	<input type="checkbox"/>	All necessary PDEs are available.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
D. Evaluation Subject as Client - Service Functionality Evaluation						
getPDEdirectory	FICS-D1	Request the directory with all attributes and no filters by sending the getPDEdirectory() message from the FICS to the evaluation tool. Make note of the author of the "REM-Test" PDE for use in <i>FICS-D4</i> .	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	FICS-D2	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the FICS to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "UID" - Filter type = "EQ" - Value = <uid of "REM-Test"> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	FICS-D3	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the FICS to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "name" - Filter type = "Like" - Value = "Recipe" 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	FICS-D4	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name, author, and createDate attributes from the FICS to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <author from REM-Test PDE> Include the createDate attribute in the getPDEdirectory() request to validate the result.	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	FICS-D5	Request the PDE directory using the following combination of filters by sending the getPDEdirectory() message with the author and gid attributes from the FICS to the evaluation tool. 1st Filter: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <string not equal to author of REM-Test> 2nd Filter: <ul style="list-style-type: none"> - Attribute = "GID" - Filter type = "EQ" - Value = <GID of "REM-Test"> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
getPDE	FICS-D6	From the FICS, request the "REM-Get1" PDE from the evaluation tool using the getPDE() message.	<input type="checkbox"/>	The correctly formed getPDE() message is received, prompting the evaluation tool to return the "REM-Get1" PDE to the equipment.		
	FICS-D7	From the FICS, request the PDEs "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDE() message requesting the multiple PDEs.	<input type="checkbox"/>	The correctly formed getPDE() message is received, prompting the evaluation tool to return the "REM-Get1" to "REM-GetN" PDEs to the equipment.		
getPDEheader	FICS-D8	From the FICS, request the header of the "REM-Get1" PDE from the evaluation tool using the getPDEheader() message.	<input type="checkbox"/>	The correctly formed getPDEheader() message is received, prompting the evaluation tool to return the "REM-Get1" header to the equipment.		
	FICS-D9	From the FICS, request the headers of the PDEs named "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDEheader() message request.	<input type="checkbox"/>	The correctly formed getPDEheader() message is received, prompting the evaluation tool to return the headers for the "REM-Get1" to "REM-GetN" PDEs to the equipment.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
sendPDE	FICS-D10	For the "REM-Delete" PDE, send the requestToSendPDE() from the FICS to the evaluation tool.	<input type="checkbox"/>	The correctly formed requestToSendPDE() message is received at the evaluation tool.		
	FICS-D11	For the "REM-Delete" PDE, send the sendPDE() from the FICS to the evaluation tool.	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDE is transferred.		
	FICS-D12	Send the getPDEdirectory() message from the FICS to the evaluation tool and verify that the "REM-Delete" PDE arrived at the evaluation tool. Include the name attribute in the getPDEdirectory() request.	<input type="checkbox"/>	The getPDEdirectory() is received, and the response includes the "REM-Delete" PDE. Note, if this is not possible to verify using the getPDEdirectory() message, it may still be possible to inspect manually on the evaluation tool.		
	FICS-D13	For the "REM-Delete" PDE, send the sendPDE() from the FICS to the evaluation tool. Note that this is an intentional repeat of <i>FICS-D11</i> .	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDE is transferred.		
	FICS-D14	Send the requestToSendPDE() message from the FICS to the evaluation tool with the PDEs "REM-Send1" to "REM-SendN" as an argument (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3).	<input type="checkbox"/>	The correctly formed requestToSendPDE() message is received at the evaluation tool.		
	FICS-D15	Send the sendPDE() message from the FICS to the evaluation tool with the PDEs "REM-Send1" to "REM-SendN" as an argument.	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDEs are transferred.		
	FICS-D16	Verify the existence of PDEs "REM-Send1" to "REM-SendN" by sending the getPDEdirectory() message requesting the name attribute from the FICS to the evaluation tool.	<input type="checkbox"/>	The getPDEdirectory() is received, and the response includes the "REM-Send1" to "REM-SendN" PDEs. Note, if this is not possible to verify using the getPDEdirectory() message, it may still be possible to inspect manually on the evaluation tool.		

Appendix C – Evaluation Procedures: EditorNodes

The following table details the procedure to execute the RaP Evaluation Method for EditorNodes. Note that rows highlighted in blue are intended to be performed on the editor, not from the evaluation tool. Ensure that the prerequisite activities in Section 7 have been performed and that Section 8 has been read and understood before beginning the evaluation.

The following columns are defined in the table:

Category – high level description of the procedure step(s)

ID – a unique identifier for the given step

Evaluation Procedure – actions to perform to execute the procedure step

Expected Result – details mechanism to verify procedure completion; a check box is provided for the use of the evaluation technician

RRI Tips – usage tips specific to the RRI to accomplish the evaluation procedure

Comments – any notes or observations regarding the execution or results of the evaluation procedure.

The appendix is designed to be printed and used to record evaluation results. However, supplemental observations must also be recorded outside of this table for use in final reporting.

Evaluation Date: _____

Evaluation Technician: _____

Equipment Supplier (make/model): _____

Comments:

Table C-1 EditorNode Evaluation Procedures

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
A. Setup						
Preparation	EDT-A1	Ensure the PDE storage area of the client is empty.	<input type="checkbox"/>	None.	Create a new storage area by updating configuration settings found in menu Options → Configuration . From the configuration dialog window, navigate to the PDE Repository tab. It is highly recommended to start a new log session, which is performed from the Session Log tab of the main window.	
Connect	EDT-A2	Establish communication and/or session with the editor. The evaluation tool must run as an FICSnode.	<input type="checkbox"/>	Check connections used: <input type="checkbox"/> SECS/HSMS (S1F13 successful) <input type="checkbox"/> RaP-XML/SOAP <input type="checkbox"/> Interface A-XML/SOAP	The simulator is started by the Start Simulator button located on the status bar at the bottom of the main window.	
Configuration	EDT-A3	Check that the necessary PDEs are on the equipment using the getPDEdirectory() message with the name attribute from the evaluation tool. (See Section 7.3)	<input type="checkbox"/>	All necessary PDEs are available	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
B. Service Functionality Evaluation						
getPDEdirectory	EDT-B1	Request the directory with all attributes and no filters by sending the getPDEdirectory() message from the evaluation tool to the editor. Make note of the author of the "REM-Test" PDE for use in <i>EDT-C5</i> .	<input type="checkbox"/>	getPDEdirectory() response provides full directory including all attributes.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EDT-B2	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the editor: <ul style="list-style-type: none"> - Attribute = "UID" - Filter type = "EQ" - Value = <uid of "REM-Test"> 	<input type="checkbox"/>	getPDEdirectory() response list includes only the "REM-Test" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EDT-B3	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the editor: <ul style="list-style-type: none"> - Attribute = "name" - Filter type = "Like" - Value = "Recipe" 	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which include the string "Recipe" in their name.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EDT-B4	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name , author , and createDate attributes from the evaluation tool to the editor: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <author from REM-Test PDE> <p>Include the author attribute in the getPDEdirectory() request to validate the result.</p>	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which match the filter, including the REM-Test PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab. Note: Be aware that dates need to be typed manually; the format is an XML Schema <i>dateTime</i> formatted string.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EDT-B5	<p>Request the PDE directory using the following combination of filters by sending the getPDEdirectory() message with the author and gid attributes from the evaluation tool to the editor.</p> <p>1st Filter:</p> <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <string not equal to author of REM-Test> <p>2nd Filter:</p> <ul style="list-style-type: none"> - Attribute = "GID" - Filter type = "EQ" - Value = <GID of "REM-Test"> 	<input type="checkbox"/>	getPDEdirectory() response list includes only PDEs which match the filter and does not include the REM-Test PDE.	<p>Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.</p> <p>Note: Be aware that dates need to be typed manually; the format is an XML Schema <i>dateTime</i> formatted string.</p>	
getPDE	EDT-B6	From the evaluation tool, request the "REM-Get1" PDE from the editor using the getPDE() message.	<input type="checkbox"/>	The editor returns the "REM-Get1" PDE in the getPDE() response, and the evaluation tool stores the PDE locally.	<p>From the RaP Requests tab, navigate to the Get PDE sub-tab.</p> <p>Note: In RRI you must first get the PDE directory before the remote PDE list will be populated in the Get PDE tab.</p>	
	EDT-B7	From the evaluation tool, request the PDEs "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDE() message requesting the multiple PDEs.	<input type="checkbox"/>	The editor returns PDEs "REM-Get1" to "REM-GetN" in the getPDE() response, and the evaluation tool stores the PDEs locally.	From the RaP Requests tab, view the Response area of the Get PDE sub-tab.	
getPDEheader	EDT-B8	From the evaluation tool, request the header of the "REM-Get1" PDE from the editor using the getPDEheader() message.	<input type="checkbox"/>	The editor returns the "REM-Get1" PDE header in the getPDEheader() response.	From the RaP Requests tab, navigate to the Get PDE Header sub-tab.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EDT-B9	From the evaluation tool, request the headers of the PDEs named "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDEheader() message request.	<input type="checkbox"/>	The editor returns the headers for PDEs named "REM-Get1" to "REM-GetN" in the getPDEheader() response.	From the RaP Requests tab, view the Response area of the Get PDE Header sub-tab.	
deletePDE	EDT-B10	Upload the PDE named "REM-Delete" to the evaluation tool by sending the getPDE() message from the evaluation tool to the editor.	<input type="checkbox"/>	The editor returns the "REM-Delete" PDE in the getPDE() response.	From the RaP Requests tab, navigate to the Get PDE sub-tab.	
	EDT-B11	Delete PDE named "REM-Delete" from the editor by sending the deletePDE() message from the evaluation tool to the editor.	<input type="checkbox"/>	The deletePDE() successfully executes with no errors. <input type="checkbox"/> PDEremoved event received by the evaluation tool with the uid of "REM-Delete"	From the RaP Requests tab, navigate to the Delete PDE sub-tab.	
	EDT-B12	Verify that the "REM-Delete" PDE is no longer resident on the editor by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the editor.	<input type="checkbox"/>	The getPDEdirectory() response shows that "REM-Delete" is not among the PDEs on the equipment.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
sendPDE	EDT-B13	For the "REM-Delete" PDE, send the requestToSendPDE() from the evaluation tool to the editor.	<input type="checkbox"/>	The requestToSendPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Request To Send PDE sub-tab.	
	EDT-B14	For the "REM-Delete" PDE, send the sendPDE() from the evaluation tool to the editor.	<input type="checkbox"/>	The sendPDE() message executes with no errors.	From the RaP Requests tab, navigate to the Send PDE sub-tab.	
	EDT-B15	Send the getPDEdirectory() message from the evaluation tool to the editor and verify that the "REM-Delete" PDE arrived at the tool. Include the name attribute in the getPDEdirectory() request.	<input type="checkbox"/>	The getPDEdirectory() response includes the "REM-Delete" PDE.	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EDT-B16	For the "REM-Delete" PDE, send the requestToSendPDE() and sendPDE() from the evaluation tool to the editor. Note that this is an intentional repeat of <i>EDT-B13 and EDT-B14</i> .	<input type="checkbox"/>	The sendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Send PDE sub-tab	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EDT-B17	Send the requestToSendPDE() message from the evaluation tool to the editor with the PDEs "REM-Send1" to "REM-Send/N" as an argument (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3).	<input type="checkbox"/>	The requestToSendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Request to Send PDE sub-tab	
	EDT-B18	Send the sendPDE() message from the evaluation tool to the editor with the PDEs "REM-Send1" to "REM-Send/N" as an argument.	<input type="checkbox"/>	The sendPDE() message executes with no errors.	From the RaP Requests tab, look at the Response area of the Send PDE sub-tab	
	EDT-B19	Verify the existence of PDEs "REM-Send1" to "REM-Send/N" by sending the getPDEdirectory() message requesting the name attribute from the evaluation tool to the editor.	<input type="checkbox"/>	The getPDEdirectory() response includes the PDEs "REM-Send1" to "REM-Send/N".	Navigate to the RaP Requests tab of the RRI and choose the Get PDE Directory sub-tab.	
	EDT-B20	Delete PDEs "REM-Send1" to "REM-Send/N" by sending the deletePDE() message from the evaluation tool to the editor referencing these PDEs.	<input type="checkbox"/>	The deletePDE() successfully executes with no errors.	From the RaP Requests tab, look at the Response area of the Delete PDE sub-tab.	
C. Evaluation Subject as Client - Setup						
Preparation	EDT-C1	If not using the pre-existing connection from the previous steps, configure the editor to connect to the evaluation tool and ensure the evaluation tool is running as an FICSnode.	<input type="checkbox"/>	None.	Run the RRI FICS Simulator. Configuration settings found in menu Options → Configuration . From the dialogue window, navigate to the SECS Configuration tab. At this time, also enter the appropriate host and port information. Note: Upon configuring the RRI, it is recommended to save your configuration for future reuse.	
Connect	EDT-C2	If not already connected, from the editor, establish communication and/or session with the evaluation tool.	<input type="checkbox"/>	Check connections used: <input type="checkbox"/> SECS/HSMS (S1F13 successful) <input type="checkbox"/> RaP-XML/SOAP <input type="checkbox"/> Interface A-XML/SOAP	The simulator is first started by the Start Simulator button located on the status bar at the bottom of the main window.	

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
Configuration	EDT-C3	Check that the necessary PDEs are on the editor using the getPDEdirectory() message with the name attribute from the evaluation tool. (See Section 7.3)	<input type="checkbox"/>	All necessary PDEs are available.		
D. Evaluation Subject as Client - Service Functionality Evaluation						
getPDEdirectory	EDT-D1	Request the directory with all attributes and no filters by sending the getPDEdirectory() message from the editor to the evaluation tool. Make note of the author of the "REM-Test" PDE for use in <i>EDT-D4</i> .	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	EDT-D2	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the editor to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "UID" - Filter type = "EQ" - Value = <uid of "REM-Test"> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	EDT-D3	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name attribute from the editor to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "name" - Filter type = "Like" - Value = "Recipe" 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
	EDT-D4	Request the PDE directory using the following filter by sending the getPDEdirectory() message requesting the name , author and createDate attributes from the editor to the evaluation tool: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <author from REM-Test PDE> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
	EDT-D5	Request the PDE directory using the following combination of filters by sending the getPDEdirectory() message with the author and gid attributes from the editor to the evaluation tool. 1st Filter: <ul style="list-style-type: none"> - Attribute = "author" - Filter type = "EQ" - Value = <string not equal to author of REM-Test> 2nd Filter: <ul style="list-style-type: none"> - Attribute = "GID" - Filter type = "EQ" - Value = <GID of "REM-Test"> 	<input type="checkbox"/>	The correctly formed getPDEdirectory() request is received.		
getPDE	EDT-D6	From the editor request the "REM-Get1" PDE from the evaluation tool using the getPDE() message.	<input type="checkbox"/>	The correctly formed getPDE() message is received, prompting the evaluation tool to return the "REM-Get1" PDE to the equipment.		
	EDT-D7	From the editor, request the PDEs "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDE() message requesting the multiple PDEs.	<input type="checkbox"/>	The correctly formed getPDE() message is received, prompting the evaluation tool to return the "REM-Get1" to "REM-GetN" PDEs to the equipment.		
getPDEheader	EDT-D8	From the editor, request the header of the "REM-Get1" PDE from the evaluation tool using the getPDEheader() message.	<input type="checkbox"/>	The correctly formed getPDEheader() message is received, prompting the evaluation tool to return the "REM-Get1" header to the equipment.		
	EDT-D9	From the editor, request the headers of the PDEs named "REM-Get1" to "REM-GetN" (where N is the highest numbered PDE created according to the instructions in Section 7.3). Perform this in a single getPDEheader() message request.	<input type="checkbox"/>	The correctly formed getPDEheader() message is received, prompting the evaluation tool to return the headers for the "REM-Get1" to "REM-GetN" PDEs to the equipment.		

Category	ID	Evaluation Procedure	<input checked="" type="checkbox"/>	Expected Result	RRI Tips	Comments
sendPDE	EDT-D10	For the "REM-Delete" PDE, send the requestToSendPDE() from the editor to the evaluation tool.	<input type="checkbox"/>	The correctly formed requestToSendPDE() message is received at the evaluation tool.		
	EDT-D11	For the "REM-Delete" PDE, send the sendPDE() from the editor to the evaluation tool.	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDE is transferred.		
	EDT-D12	Send the getPDEdirectory() message from the editor to the evaluation tool and verify that the "REM-Delete" PDE arrived at the evaluation tool. Include the name attribute in the getPDEdirectory() request.	<input type="checkbox"/>	The getPDEdirectory() is received, and the response includes the "REM-Delete" PDE. Note, if this is not possible to verify using the getPDEdirectory() message, it may still be possible to inspect manually on the evaluation tool.		
	EDT-D13	For the "REM-Delete" PDE, send the sendPDE() from the editor to the evaluation tool. Note that this is an intentional repeat of <i>EDT-D11</i> .	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDE is transferred.		
	EDT-D14	Send the requestToSendPDE() message from the editor to the evaluation tool with the PDEs "REM-Send1" to "REM-SendN" as an argument (where <i>N</i> is the highest numbered PDE created according to the instructions in Section 7.3).	<input type="checkbox"/>	The correctly formed requestToSendPDE() message is received at the evaluation tool.		
	EDT-D15	Send the sendPDE() message from the editor to the evaluation tool with the PDEs "REM-Send1" to "REM-SendN" as an argument.	<input type="checkbox"/>	The correctly formed sendPDE() message is received at the evaluation tool, and the PDEs are transferred.		
	EDT-D16	Verify the existence of PDEs "REM-Send1" to "REM-SendN" by sending the getPDEdirectory() message requesting the name attribute from the editor to the evaluation tool.	<input type="checkbox"/>	The getPDEdirectory() is received, and the response includes the "REM-Send1" to "REM-SendN" PDEs. Note, if this is not possible to verify using the getPDEdirectory() message, it may still be possible to inspect manually on the evaluation tool.		

**International SEMATECH Manufacturing Initiative
Technology Transfer
2706 Montopolis Drive
Austin, TX 78741**

**<http://ismi.sematech.org>
e-mail: info@sematech.org**