ISE / RASSP

Design Guide
Reports Supplement

Version 1
DECEMBER 1996

Ascent Logic Corporation

The information in this publication is subject to change without notice. Other product names mentioned in this document are trademarks or registered trademarks of their respective owners and are used for identification purposes only.

Restricted Rights Legend

Ascent Logic Corporation
180 Rose Orchard Way, #200
San Jose, California 95134

P/N 101107DGR

Contents

OUTPUTSECTIONSREPORTBLOCKS C-1

Preface

This manual describes the reports developed under RASSP and provided in the Integrated System Engineering (ISE) environment in order to produce formal documentation from an RDD-100® database developed using the ISE schema.

The preface includes these sections:

SCOPE

The Design Guide Report Writer application developed under RASSP for the ISE schema produces selected MIL-STD-490A/DOD-STD-2167A and MIL-STD-498 documentation from system engineering data created in RDD-100 using the RASSP schema and supporting methodology. This supplement gives an overview of the reports and Document Management Facility (DMF) templates comprising the application, describes the element types, relationships, and attributes used to produce each specification, and explains how to execute the Design Guide reports.

DOCUMENT PURPOSE

This document is intended to assist the user in producing documents from an RDD-100 database. It describes the structure of the Design Guide reports and provides the steps necessary to populate the DMF, link the DMF to system engineering information in other RDD-100 facilities, and finally to execute the Design Guide reports that create the actual document.

The Design Guide reports are used to produce MIL-STD documents. However, the modularization of the Design Guide reports and associated files supports customizing of the specifications, either to the user's own needs or to alternate documentation standards.

The Design Guide reports allow users to generate documents at any stage in the design process. Working early in the process with the initial data, users can generate a partial specification. As engineering work progresses, new material can be added.

The reports also enable you to classify RDD-100 elements and report sections, generate Project Unique Identification numbers, allow space in the specification for externally prepared encapsulated PostScript or Interleaf figures and tables, and automatically create subsections of the document based on the relationships of RDD-100 elements.

DOCUMENT ORGANIZATION

Instructions for preparing the RDD-100 System Description Database (SDD) and running the Design Guide reports are described in the chapters and appendices described below.

This supplement is a combination of user's guide and reference manual for the Design Guide reports. Chapters 1 through 6 make up the user's guide, while Appendices A through C contain reference information:

Chapter 1, "Specification Generation Overview."	Introduces the RDD-100 Document Management Facility (DMF) and associated reports, identifies the specifications available, and lists the directory location and filenames for each report and document template provided with this application.

Chapter 2, “Document Generation Using Templates.”	Describes the general procedures for producing a 490A/2167A or 498 specification using the Design Guide’s report generation capabilities. These procedures include: importing the engineering data and DMF elements; creating specification subsections; and printing the report.
Chapter 3, “Executing Design Guide Reports.”	Gives instructions for actually executing the three reports that support the Design Guide: Autocreate Subsections, Project Unique ID, and Output Sections.
Chapter 4, “Use of DMF by Design Guide Reports.”	Describes the element types that make up the DMF and how they are used by the Autocreate Subsections Report and the Output Sections Report.
Chapter 5, “Use of Engineering Elements by Design Guide Reports.”	Describes the system engineering element types, attributes, and relationships used by the Autocreate Subsections Report and the Output Sections Report.
Chapter 6, “Customization.”	Describes how to customize Design Guide reports and Document Section Templates.
Appendix A, “Document Sections Templates”	Describes the Document Sections Templates provided with the Design Guide Type R reports.
Appendix B, “Autocreate Subsections Report Blocks”	Describes the Autocreate Subsections *ReportBlock* elements.
Appendix C, “Output Sections Report Blocks”	Describes the Output Sections *ReportBlock* elements.

RELATED DOCUMENTS

This supplement should be used in conjunction with the following set of related documentation:

- ❑ *RDD-100 User's Manual for the ISE (RASSP) Schema*
- ❑ *RDD-100 Release 4.1.1 On-line Release Notes*
- ❑ *RDD-100 Release 4.1.1 On-line Help*

FONT CONVENTIONS

Fonts have meanings in this manual as described below:

FONT	EXAMPLE	DESCRIPTION
Helvetica Plain	Creation Date	Attribute name
Helvetica Italic	*Source*	Element type name
Helvetica Bold	**inputs**	Relationship name
Helvetica Bold Italic	***File > save image***	Menu option or "button" command
`Courier Plain`	`Database.rdt`	File and directory names, operating system commands, RDD-100 messages, text that you type, and predefined elements (i.e., *ReportBlock* names)
Times Italic	*filename*	Generic name where you substitute specific text for a variable

Chapter

1

Specification Generation Overview

This chapter introduces the RDD-100 Document Management Facility (DMF) and identifies the 490A/2167A and 498 specifications supported by the Design Guide Type R Reports.

It also provides a brief description of the Design Guide Reports and lists the file names for each of the Design Guide Reports and document sections templates provided with this application.

The chapter includes these sections:

INTRODUCTION

Over the course of an engineering development cycle—ideally, over the entire life cycle of the system—the RDD-100 user incrementally builds an integrated body of design data that is stored in the System Description Database (SDD). Over that same time period, it will be necessary to print that SDD information for a variety of purposes and recipients. Thus, sometimes the same data must be printed in different documents; and sometimes the same document will, as time progresses, report on an expanding set of data.

This situation requires a flexible reporting tool that on the one hand supports standard documents and on the other hand supports the easy generation of specialized documents that may be unique to a user organization. In RDD-100, this combination of standardization and flexibility is provided by the Design Guide Reports, supported by the Document Management Facility.

DOCUMENT MANAGEMENT FACILITY (DMF)

To produce documents using the Design Guide Reports, it is necessary first to understand the Document Management Facility (DMF). The DMF contains seven element types, which are described in detail in later chapters. At this point, however, it is sufficient to focus on two of those element types: the *ReportDocument* and the *Section*.

The Design Guide Reports take advantage of the fact that it is both possible and convenient to think of the structure of a document separately from its content.

- ❑ In general, the content of the document is supplied by the engineering information that is contained in the System Description Database (the descriptions of and relationships between requirements, functions, components, and so on).
- ❑ The document structure, on the other hand, is defined by the *ReportDocument* and the *Section* elements in the Document Management Facility (DMF).

These elements in the DMF serve, in effect, as a table of contents for the document, with references that point to the engineering data that will go in each section. To print a document, merely link the *Sections* to the appropriate elements in the engineering data, confirm the selection of autocreate and print *ReportBlocks* for each Section, create any additional subsections manually or by using the Autocreate Subsections Report (one of the Design Guide Reports), and then print the document using the Output Sections Report..

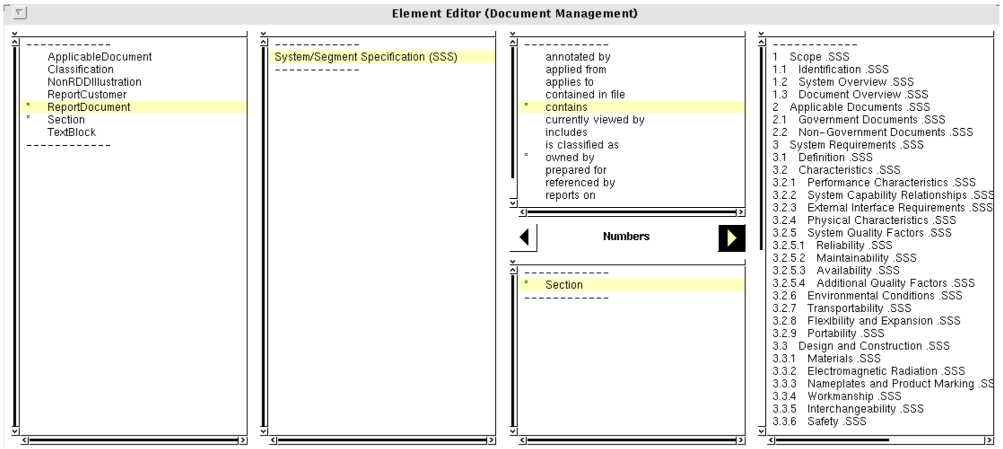
Users may approach the creation of documents in a variety of ways. The way users choose will depend upon the type of document needed and their level of knowledge of the RDD-100 Report Writer.

- ❑ At one end of the scale, users can use the templates as supplied by Ascent Logic Corporation. These templates provide document structures for the standard 490A/2167A and 498 specifications. The templates may be simply imported into a user image and used without modification.
- ❑ At the other end of the scale, users can create completely original documents using new DMF elements and user-written *ReportBlocks*. In this case, users need familiarity with the RDD-100 Report Writer to write the new *ReportBlocks*.

In between these two extremes, a wide range of customization possibilities exists. Users can modify the DMF elements, the *ReportBlocks*, or both. The amount of customization they can do is limited only by their documentation needs and by their skill level with the Report Writer (if new *ReportBlocks* are to be created). The simplest type of customization would be to rearrange existing *Sections* and add new *Sections*. More sophisticated modifications would involve the use of different *ReportBlocks* or the creation of new *ReportBlocks*.

For the purpose of this introduction, the example uses the template for the System/Segment Specification (SSS) supplied by Ascent Logic Corporation. However, it is important to remember that the DMF elements and *ReportBlocks* in that template were initially created as an “original” document which we now make available as a “standard template”. The elements may be modified as necessary to support program needs, and the modified templates can then serve as a “standard” document that can be imported and used by anyone in your organization.

When the DMF elements are created or imported, they provide a high-level outline of a particular document. For example, Figure 1-1 shows how the DMF elements look in the Element Editor immediately after importing the file named “DGR490-SSS-ASpecDoc.rdt”. This file, which is supplied by Ascent Logic Corporation, contains the base set of DMF elements used to produce a 490A type of SSS.



Element Editor (Document Management)

```
------------
     ApplicableDocument
     Classification
     NonRDDIllustration
     ReportCustomer
*    ReportDocument
*    Section
     TextBlock
------------
```

```
------------
System/Segment Specification (SSS)
------------
```

```
------------
     annotated by
     applied from
     applies to
     contained in file
*    contains
     currently viewed by
     includes
     is classified as
*    owned by
     prepared for
     referenced by
     reports on
```

```
------------
*    Section
------------
```

```
------------
1   Scope .SSS
1.1   Identification .SSS
1.2   System Overview .SSS
1.3   Document Overview .SSS
2   Applicable Documents .SSS
2.1   Government Documents .SSS
2.2   Non-Government Documents .SSS
3   System Requirements .SSS
3.1   Definition .SSS
3.2   Characteristics .SSS
3.2.1   Performance Characteristics .SSS
3.2.2   System Capability Relationships .SSS
3.2.3   External Interface Requirements .SSS
3.2.4   Physical Characteristics .SSS
3.2.5   System Quality Factors .SSS
3.2.5.1   Reliability .SSS
3.2.5.2   Maintainability .SSS
3.2.5.3   Availability .SSS
3.2.5.4   Additional Quality Factors .SSS
3.2.6   Environmental Conditions .SSS
3.2.7   Transportability .SSS
3.2.8   Flexibility and Expansion .SSS
3.2.9   Portability .SSS
3.3   Design and Construction .SSS
3.3.1   Materials .SSS
3.3.2   Electromagnetic Radiation .SSS
3.3.3   Nameplates and Product Marking .SS
3.3.4   Workmanship .SSS
3.3.5   Interchangeability .SSS
3.3.6   Safety .SSS
```

FIGURE 1-1 DMF Elements for SSS

The document structure is controlled by two types of elements, a *ReportDocument* and a number of *Sections* that are linked to it:

- For each different document, there is one element of type *ReportDocument*. In this example, which shows the elements used for an SSS, the *ReportDocument* element is named “System/Segment Specification (SSS)”.
- Associated with a *ReportDocument* element is a set of elements of type *Section* that are linked by the **contains** relationship (that is, a *ReportDocument* **contains** *Sections*). These *Sections* are named and numbered to correspond to the major sections

of the document represented by the *ReportDocument*. Note that the DMF element section structure is flat. Each section is linked directly to the *ReportDocument* element with which it is associated, as is illustrated in Figure 1-2 (only part of the view is shown)

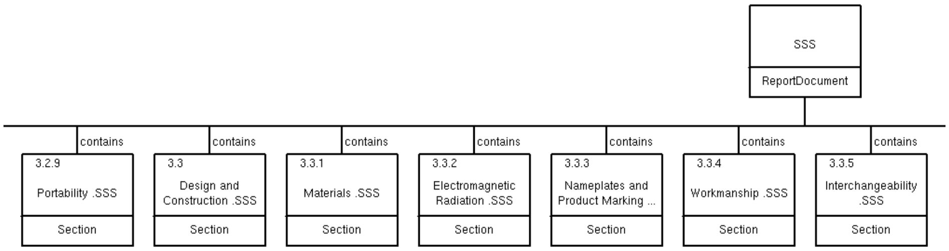


FIGURE 1-2 Custom Hierarchy View of SSS Elements in DMF

DMF elements for the standard specifications are delivered to the RDD-100 user in .rdt files called Document Sections Templates. A Document Sections Template is provided for each type of 490A/2167A and 498 specification that is supported (see the next section for supported specifications). Each contains one *ReportDocument* and the *Sections* that the *ReportDocument* **contains**. In addition, in separate files Ascent Logic Corporation provides a set of *Category* elements appropriate to the different specifications. If necessary, these templates may be tailored by the user organization to fit their particular environment.

Although the primary function of the DMF element types is to define the report structure, several element types also can be used to contribute content (generally administrative) to the report.

- *ApplicableDocument* provides information used to print the Applicable Documents section (i.e., Section 2).
- *Classification* defines the classification markings that are to be printed.
- *NonRDDIllustration* identifies graphics or space for graphics not stored in RDD-100 that are to be included in a document.
- *ReportCustomer* identifies the recipient of the report for printing on the title page.
- *TextBlock* stores arbitrary text that is to be printed in the document.

The chapters that follow in this document provide both a description of the steps required to generate documents and the reference information needed if a user organization wishes to tailor any of those documents to their specific needs.

SPECIFICATIONS SUPPORTED

The 490A/2167A and 498 specifications and documents supported by the Design Guide Type R reports are:

MIL-STD-490A/DOD-STD-2167A

- ❑ *System/Segment Specification (SSS or A Spec.)*
  DI-CMAN-80008A
- ❑ *Prime Item Development Specification (B1 Spec.)*
  MIL-STD-490A
- ❑ *Critical Item Development Specification (B2 Spec.)*
  MIL-STD-490A
- ❑ *System/Segment Design Document (SSDD)*
  DI-CMAN-80534
- ❑ *Software Requirements Specification (SRS)*
  DI-MCCR-80025A
- ❑ *Interface Requirements Specification (IRS)*
  DI-MCCR-80026A

MIL-STD-498

- ❑ *System/Subsystem Specification (SSS)*
  DI-IPSC-81431
- ❑ *System/Subsystem Design Description (SSDD)*
  DI-IPSC-83421
- ❑ *Software Requirements Specification (SRS)*
  DI-IPSC-81433
- ❑ *Interface Requirements Specification (IRS)*
  DI-IPSC-81434

DESIGN GUIDE REPORTS

Ascent Logic Corporation provides three reports that are used in the document generation process. Instructions for executing these reports are presented in Chapter 3, “Executing Design Guide Reports”.

- **PROJECT UNIQUE ID (PUID) NUMBERING REPORT.** Assigns unique identifiers to the engineering elements that are to be reported on. The identifier is stored in the attribute called Project Unique ID of the engineering elements.
- **AUTOCREATE SUBSECTIONS REPORT.** Creates subsections for a *ReportDocument*. This report provides the detailed linkage between the engineering data and the DMF elements so that the specification may then be printed using the Output Sections Report.
- **OUTPUT SECTIONS REPORT.** Extracts and formats the information to be printed in each individual section of a desired document (defined by a *ReportDocument*). Output is written to a file in PostScript, Interleaf, or ASCII format and is suitable for printing or for on-line display.

REPORT AND DOCUMENT TEMPLATE FILES

The Design Guide Reports, Document Sections Templates, and predefined *Categories* are provided as .rpt and .rdt files. They should be located in the following directory:

UNIX PLATFORMS /*<RDD installation directory>*`/RDD-100-R4.1.1/SupportInfo/ISE/DGR`

The following table lists the filenames of the reports, document sections templates, and category files used for the Design Guide Type R. Note that the Autocreate Subsections and Output Sections reports are contained in a single file. Different specifications are produced by using the same reports but with a different document sections template.

TABLE 1-1 Specification-Related Files for Design Guide Type R

UNIX FILE	CONTENTS
`DGR-Reports.rpt`	Autocreate Subsections and Output Sections reports
`DGR-ProjectUniqueID.rpt`	Project Unique ID Numbering report
`DGR490-SSS-ASpecDoc.rdt`	SSS Document Sections Template (490A/2167A)
`DGR490-SSDDDoc.rdt`	SSDD Document Sections Template (490A/2167A)
`DGR490-SRSDoc.rdt`	SRS Document Sections Template (490A/2167A)
`DGR490-IRSDoc.rdt`	IRS Document Sections Template (490A/2167A)
`DGR490-B1Doc.rdt`	B1 Document Sections Template (490A/2167A)
`DGR490-B2Doc.rdt`	B2 Document Sections Template (490A/2167A)
`DGR490-SSSCategories.rdt`	SSS Categories (490A/2167A)
`DGR490-SSDDCategories.rdt`	SSDD Categories (490A/2167A)
`DGR490-SRSCategories.rdt`	SRS Categories (490A/2167A)

TABLE 1-1 Specification-Related Files for Design Guide Types (Continued)

UNIX FILE	CONTENTS
`DGR490-IRSCategories.rdt`	IRS Categories (490A/2167A)
`DGR490-B1Categories.rdt`	B1 Categories (490A/2167A)
`DGR490-B2Categories.rdt`	B2 Categories (490A/2167A)
`DGR498-SSSDoc.rdt`	SSS Document Sections Template (498)
`DGR498-SSDDDoc.rdt`	SSDD Document Sections Template (498)
`DGR498-SRSDoc.rdt`	SRS Document Sections Template (498)
`DGR498-IRSDoc.rdt`	IRS Document Sections Template (498)
`DGR498-SSSCategories.rdt`	SSS Categories (498)
`DGR498-SSDDCategories.rdt`	SSDD Categories (498)
`DGR498-SRSCategories.rdt`	SRS Categories (498)
`DGR498-IRSCategories.rdt`	IRS Categories (498)

Chapter

2

Document Generation Using Templates

This chapter describes the general procedure for producing a 490A/2167A or 498 document for the first time using the Design Guide report generation capabilities. The procedure involves three main steps, which are described by the chapter's three sections:

IMPORT THE ENGINEERING DATA AND DMF ELEMENTS

IMPORT THE ENGINEERING DATA

The first step is to import any existing engineering data into the image from which the document will be generated. This image must contain the ISE schema, plus any user extensions. You might then run the appropriate consistency checks to make certain that the engineering data is at the desired level of completion.

The Design Guide reports print information only on those elements that are pertinent to a particular document. Chapter 5, “Use of Engineering Elements by Design Guide Reports”, contains a general discussion of the engineering elements used by the reports.The schema table in Appendix A, ”Document Sections Templates”, indicate what elements are associated with each specification. All engineering elements that provide content to a document using the Design Guide reports should, at minimum, have values assigned to their Description and Project Unique ID attributes. If your organization has customized the reports, additional information may also be required.

The Project Unique IDs Numbering Report assigns values to the Project Unique ID attribute of each element. The identifier assigned by the report consists of a prefix that combines an optional project identifier created by the user and an element type abbreviation, followed by an integer. The elements of each element type are numbered in separate numerical sequences. As new elements are added to the database, the report can be re-executed to assign project identifiers to the new data. Chapter 3, “Executing Design Guide Reports” tells how to run it.

IMPORT THE DMF ELEMENTS

Besides the engineering data, the image must also contain the set of DMF elements that provides the structure for the desired specification. Candidate structures have been provided in Document Sections Template files (for example, `DGR490-SSS-ASpecDoc.rdt`); you must either import a template file into the RDD-100 image that contains the engineering data or define a structure in the DMF.

Each Document Sections Template file consists of a single *ReportDocument* element that **contains** many *Section* elements. For example, the template for the 490A type of SSS has 56 sections.

The name of a *Section* element in a template file has this form:

title .specid

In the example above, *title* is the value of the element's Title attribute and *specid* is an acronym that identifies the specification (such as "SSS"). A space is interposed between *title* and the period.

For example, the *Section* element for section 1.2 of the SSS has the character string "System Overview" in its Title attribute, and its Number attribute is 1.2. Its name is "System Overview .SSS". When viewing *Sections* in the Element Editor of an image that contains multiple document definitions, this acronym can be used as a filter to view only those *Sections* for a particular document.

CREATE SPECIFICATION SUBSECTIONS

The next step is to create the SDD dependent subsections for the report. This involves two activities:

- ❑ Assign attribute values and relationship targets for the DMF elements.
- ❑ Execute the Autocreate Subsections Report.

Subsections also may be created manually instead of using the Autocreate Subsections Report. This is described later in this chapter.

ASSIGN ATTRIBUTE VALUES AND RELATIONSHIPS FOR DMF ELEMENTS

The following list describes the set of information that users must provide for the DMF elements before executing the Autocreate Subsections Report. See Chapter 4, “Use of DMF by Design Guide Reports” for a complete definition of each of the attributes and relationships.

ReportDocument The Document Type attribute must have one of the following values: SSS, B1, B2, SSDD, IRS, or SRS. If this attribute is not defined, the report prints an error message and terminates.

The Abbreviation attribute must have a value, preferably a unique value appropriate for the document (such as SSS). If this attribute is not defined, the report prints an error message and terminates. This attribute value should match the abbreviation portion of the suffix (that is, specid) of any existing *Section* contained by the *ReportDocument*. To modify an existing *ReportDocuments* Abbreviation attribute and its *Sections* suffix, see “Creating Multiple Documents of the Same Type” on page 3-8.

The Format Standard attribute should be set to the standard being followed, either 490A/2167A or 498. If this attribute is not defined, 490A/2167A will be assumed by any *ReportBlocks* requiring this information.

The relationship **reports on** must have a target *Component*. If this relationship is not defined, the report prints an error message and terminates.

The relationship **contains** should have as targets all of the *Sections* which are to be included in the document. *Sections* that are not targets will be ignored (no subsections will be created).

Section The Number attribute must be a hierarchical number that identifies the position of the *Section* in the specification. If the Number attribute of a *Section* is blank, the Autocreate Subsections Report assigns it the value “`No.Number.Assigned`” and stops processing that *Section* (creates no subsections).

The Autocreate/Output Subsections Blocks attribute should contain the name of the Autocreate *ReportBlock* that will be used to create its subsections and the name of the Output *ReportBlock* that will be used to print those subsections. If line 1 of the attribute does not contain the name of an Autocreate *ReportBlock*, the *ReportBlock* named “`>@ generic`” will be used to generate subsections.

If the name of an Output *ReportBlock* is to be assigned to the Report Block Name attribute of a subsection, that name must be included as the second line of the Autocreate/Output Subsections Blocks attribute in the parent *Section*.

A starting element for the Autocreate *ReportBlock* should be present in the database.

- If no Autocreate *ReportBlock* is specified in the attribute Autocreate/Output Subsections Blocks, any engineering elements that are to be used in creating subsections by the default Autocreate *ReportBlock* must be targets of the **uses** relationship from the *Section*.
- If an Autocreate *ReportBlock* is specified in the attribute Autocreate/Output Subsections Blocks, the starting element needed by that *ReportBlock* must be in the engineering data and must have the relationships as specified in the description of the *ReportBlock* (see Appendix B, ”Autocreate Subsections Report Blocks”, for descriptions of all the Autocreate *ReportBlocks*).

If these relationships are absent, the report will execute but will not create the intended subsections.

TextBlock Text that is to appear in a specification but which is not found in the engineering elements. Data may be entered into the Description of a *TextBlock* element. If *TextBlocks* are to appear as numbered subsections in the document, those *TextBlocks* should then be made the target of the **uses** relationship from the desired *Section* and the Autocreate *ReportBlock* for that *Section* should be set to “`>@ TextBlocks`”.

For appropriate *Sections* in the Document Sections Templates, the Autocreate/Output Subsections Blocks attribute has been set to suggested Autocreate and Output *ReportBlocks*. Users may wish to use different *ReportBlocks* to autocreate sections, or they may wish to tailor *ReportBlocks* to suit particular reporting requirements of their organization or program. Appendices B and C describe the actions of the Autocreate and Output *ReportBlocks*, respectively.

EXECUTE THE AUTOCREATE SUBSECTIONS REPORT

Once the engineering data is available and the DMF elements have been assigned the values as described above, the next step is to execute the Autocreate Subsections Report. This report performs two important tasks:

❑ It creates the lower-level subsections.

❑ It inserts printing instructions in the new subsections.

Detailed information about how to execute the Autocreate Subsections Report is contained in Chapter 3, “Executing Design Guide Reports”.

REPORTPROCESSING

When users run the Autocreate Subsections Report, they select a particular *ReportDocument* for the report to process. The Autocreate Subsections Report processes all the *Sections* that are targets of the **contains** relationship from that *ReportDocument*, sorting the *Sections* hierarchically by the Number attribute. The Autocreate/Output Subsections

Blocks attribute of each *Section* contains the instructions (in the form of references to elements of type *ReportBlock*) for creating its subsections.

The *ReportBlocks* executed by the Autocreate Subsections Report are referred to as Autocreate *ReportBlocks*. They are easily identified because the name of each begins with the two characters “`>@`”. Appendix B, ”Autocreate Subsections Report Blocks”, contains descriptions of the pre-defined Autocreate *ReportBlocks*.

For each *Section*, the report performs the following actions, depending upon the value(s) in the Autocreate/Output Subsections Blocks attribute:

- ❑ If an Autocreate *ReportBlock* is specified, the report executes the instructions contained in that *ReportBlock* to create subsections for the parent *Section*. Depending upon the contents of the *ReportBlock* and the engineering data, multiple levels of subsections may be created. If the *ReportBlock* requires targets for relationships that have not been specified or are not in the database, some or all of the expected subsections will not be created.

  If the Autocreate/Output Subsections Blocks attribute also contains the name of an Output *ReportBlock* (that is, its name begins with the character string “`>Print`”), the report inserts the name of that *ReportBlock* in the Report Block Name attribute of each subsection it creates. Note: Some Autocreate *ReportBlocks* override this direction for certain autocreated sections and insert pre-defined names. These special cases are noted in Appendix B.

- ❑ If the Autocreate/Output Subsections Blocks attribute is empty, the *ReportBlock* named “`>@ generic`” is executed. It creates a subsection for each engineering element that is the target of the **uses** relationship from the *Section*.
- ❑ If the value on line 1 of the Autocreate/Output Subsections Blocks attribute is the name of an Output *ReportBlock* (rather than an Autocreate ReportBlock), the Autocreate Subsections Report will generate error messages when it attempts to process that *Section*.

RESULTS OF REPORT PROCESSING

When the Autocreate Subsections Report executes successfully, one or more new *Section* elements are added to the database as targets of the **contains** relationship from the *ReportDocument*. Each autocreated subsection has the following characteristics:

- ❑ Its name has the following format:

  ```
  title .abb@
  ```

  The first part of the name (`title`) is set to the value of the Title attribute of the engineering element that is used to produce the contents of the new *Section*. If that attribute is empty, the name of the element is used. This name is followed by a space and a period. A suffix is then added, containing the value in the Abbreviation attribute of the *ReportDocument* (`abb`) followed by the character “@”. If the resulting name is not unique, an additional suffix is added. The additional suffix consists of a period followed by a unique integer. The “@” suffix makes it easy to use a filter in the Element Editor to list either all the autocreated or all the non-autocreated *Sections*.

In some cases, the report creates a standard *Section* that is not associated with a particular engineering element (for example, section 3.X.2 of an IRS). In these cases, the name is set by the Autocreate *ReportBlock* to a standard pre-defined name appended with the appropriate suffix as described above.

- ❑ It is a target of the **contains** relationship from the *ReportDocument*.
- ❑ Its Number attribute is set to a hierarchical number based on the Number of the parent *Section*. If a *ReportBlock* traverses a hierarchy of elements, creating subsections for several levels of elements, the created *Sections* will form a hierarchy by number beneath the parent *Section*.
- ❑ Its Report Block Name attribute may contain the name of an Output *ReportBlock* inserted by the Autocreate *ReportBlock* that created the *Section*.
- ❑ Depending on the Autocreate *ReportBlock* that created the *Section*, that *Section* may be linked to an engineering element by the **describes** relationship.

If you forget to set the proper Autocreate *ReportBlock* name, the wrong set of *Section* elements may be created. If this happens, delete those new *Section* elements, set the attribute to the desired *ReportBlock*, and re-run the Autocreate Subsections Report selecting only the *Section* which needs autocreation.

Step-by-step instructions for executing the Autocreate Subsections Report are contained in Chapter 3, “Executing Design Guide Reports”.

MANUAL CREATION OF SUBSECTIONS

Subsections may also be created manually, either instead of or in addition to using the Autocreate Subsections Report. New top-level sections may be created in the same way. When there are just a few new sections to add, users normally create them manually. To add a subsection, the user only needs to know the name and placement of the subsection and the engineering elements that will be printed within it.

The following checklist indicates the information that should be added to the database to create a meaningful new *Section*:

- ❑ Create a new *Section* element as a target of the **contains** relationship from the desired *ReportDocument*. Use the value that is in the Abbreviation attribute of the *ReportDocument* as a suffix for the name of the *Section*.
- ❑ Assign values to the attributes of the *Section* as follows:
  - ■ Assign a value to the Number attribute so that the *Section* will appear in the proper location in the document. For example, if the new *Section* is to be a subsection of section 3.5, its Number should be 3.5*x*, where *x* is an integer that specifies the next highest unused number.
  - ■ Enter the name of the desired Output *ReportBlock* in the Report Block Name attribute of the *Section*. If that attribute is left blank, the Output Sections Report executes the *ReportBlock* named “`>Print Description`” when processing the section.
  - ■ In the Title attribute, enter the name of the section as it should be printed in

the document. If the Title attribute is blank, the Output Sections Report will print the element name as the title.

- If you wish to define text to be printed in the document immediately beneath the section heading, enter that text in the Description attribute of the Section. (Alternatively, a *TextBlock* can be used; see next bullet.)

❑ Assign targets for the following relationships from the *Section*:

- If a classification is to be printed for the *Section*, select the appropriate *Classification* element as the target of the **classified by** relationship.
- If text with different classifications is to be printed in un-numbered paragraphs of the *Section*, place each paragraph of text in an element of type *TextBlock* that is the target of the **described by** relationship from the *Section*. Assign the desired classifications to the *TextBlock* elements. Assign the Number attribute of each *TextBlock* to a number indicating the order in which to print the text.
- If engineering elements are to be printed as part of the *Section*, make those elements targets of the **described by** relationship from the *Section*.

PRINT THE REPORT

The third and final step is to create (and optionally print) a printable document file that includes classification markings; space for externally generated graphics (if any); and a title page, table of contents, list of figures, and list of tables. To produce this file two activities are required:

❑ Verify that the DMF elements have the needed attribute values and relationships.

❑ Execute the Output Sections Report.

VERIFY DMF ATTRIBUTES AND RELATIONSHIPS

To produce a complete specification, the Output Section Report requires the following set of information for the DMF elements. If the Autocreate Subsections Report was used to create the subsections, the information for the *ReportDocument* and *Section* elements should be complete. See Appendix C, ”Output Sections Report Blocks”, for a complete definition of each of the attributes and relationships.

ReportDocument

The Document Type attribute must have one of the following values: SSS, B1, B2, SSDD, IRS, or SRS. If this attribute is not defined, the report prints an error message and terminates.

The Format Standard attribute should be set to the standard being followed, either 490A/2167A or 498. If this attribute is not defined, 490A/2167A will be assumed.

The **reports on** relationship must have a target *Component*. If this relationship is not defined, the report prints an error message and terminates.

All *Sections* that are to be printed must be a target of the **contains** relationship from the *ReportDocument*.

The following attributes provide information that is printed in the header of each page and/or on the title page and therefore should be assigned the appropriate values: CDRL Number, Contract Number, Contractor Address, Contractor Name, Document Control Number, and Document Type.

Section

The Number attribute must have a value. A *Section* without a Number is not printed.

The Title attribute provides the heading that will be printed for the *Section*. If the Title attribute is blank, the name of the *Section* is printed as the section heading.

The Report Block Name attribute of the *Section* specifies the name of the Output *ReportBlock* used to print that *Section*. The Name of an Output *ReportBlock* always begins with the character string “`>Print`”. If a *ReportBlock* specified in the Report Block Name attribute of a *Section* is not in the image, the report ends processing of the *Section* and a diagnostic is generated. If an Output *ReportBlock* is executed against a database in which some of the attributes and relationships required by the *ReportBlock* are not specified, the *Section* is printed with only partial information.

TextBlock

Any *TextBlock* that is the target of the **describes** relationship from a *Section* should have a value in its Description attribute. That value is printed as an unnumbered paragraph in the *Section*.

Applicable-Document

The following attributes should have values: Document Number, Government, and Publication Type.

A *ReportDocument* target must be specified for the **applies to** relationship.

Classification

The following attributes should have values: Abbreviation, Rank, Title, and Usage.

Targets must be specified for the **classifies** relationship.

NonRDD-Illustration

If the specification is to directly include or allow room for graphics produced outside of RDD-100, the following attributes should have values: Document Type, IllustrationType, Title, and either Number of Pages or EPS Picture Height, EPS Picture Width, and Include File.

Targets must be specified for the **illustrates** relationship.

ReportCustomer

The following attributes should have values: Address and Department Code.

A *ReportDocument* target must be specified for the **receives** relationship in order for the *ReportCustomer* information to be printed on the title page.

EXECUTE THE OUTPUT SECTIONS REPORT

The Output Sections Report produces a document file as its output. If the output format is PostScript or ASCII, it is printable as produced. If the output format is Interleaf, additional processing in Interleaf is required.

When executing the Output Sections Report, users must select a particular *ReportDocument* for the report to process. The report then processes all the *Sections* that are targets of the **contains** relationship from that *ReportDocument*, taking its printing instructions from the Output *ReportBlock* referenced in the Report Block Name attribute of each *Section*.

For each *Section*, the report performs the following actions, depending upon the value(s) in that attribute:

- If the Report Block Name attribute of the *Section* contains the name of an Output *ReportBlock*, the report executes that *ReportBlock* and no further action is performed for that *Section*.
- If Report Block Name is blank, the report uses the Output *ReportBlock* named “`>Print Description`”. This results in the following output for the *Section* in the order listed:
    - The Number and Title of the *Section* is printed as a heading.
    - The Description attribute of the *Section* is printed (if it has a value).
    - If the *Section* has *TextBlock* targets for the **describes** relationship, the Description attribute of each target *TextBlock* is printed as a separate, unnumbered paragraph.
    - If the *Section* has targets for the **describes** relationship that are not *TextBlocks*, the target's Description is generally printed as a separate, unnumbered paragraph appended with any Project Unique ID value for that element. *Requirements* elements have special print rules as described in “Standard Element Print Information” on page C-3.

Step-by-step instructions for executing the Output Sections Report are contained in Chapter 3, “Executing Design Guide Reports”.

Chapter

3

Executing Design Guide Reports

This chapter describes how to execute the Design Guide Reports. It includes these sections:

STANDARD PROCEDURE FOR EXECUTING REPORTS

⚠ **CAUTION:** Because the Autocreate Subsections Report modifies your database, always make a copy of your data and save your image before executing that report. This protects you against any possible errors by making it easy to return to the pre-execution state.

In RDD-100, reports can be executed by selecting either ***Print>external report*** or ***Print>internal report*** from the RDD-100 Main Menu. (You may also use the Command Reader to execute reports; see “Executing Reports Using the Command Reader” on page 3-7 for details.)

❑ Use ***Print>internal report*** to execute reports that have been imported into the image. When this option is selected, a selection window opens to display the *ReportMainBlocks* that are available for execution. The Autocreate Subsections and Output Sections Reports will be on this list if the `DGR-Reports.rpt` file was imported. Select the desired report and ***Accept*** the window to run the report.

❑ Use ***Print>external report*** to execute reports stored in an External Report Library. (External libraries are defined in the User Preferences Window.) When the submenu listing External Report Libraries appears, select the appropriate library, then ***DGR Reports***. When the menu listing the Design Guide Reports is displayed, select the name of the desired report (either `Autocreate Subsections` or `Output Sections`). This imports and then executes the report.

After you select an internal or external report, the Report Output Setup Window is displayed. Set the print parameters as desired:

❑ When executing the Autocreate Subsections Report, select either the ASCII or PostScript formatting option and close the window with the ***File*** rather than the ***Print*** option. The output file (which contains diagnostics if any errors were found plus information on the *Sections* created during report execution) can then be easily viewed online with an ASCII text editor or PostScript previewer.

❑ When executing the Output Sections Report, select either PostScript or Interleaf output formats and close the window with either the ***Print*** or the ***File*** option. For Interleaf output, always select ***File***. For PostScript output, you may select ***Print*** or ***File***.

After the Report Output Setup Window closes, the processes for executing the two reports differ.

EXECUTING THE AUTOCREATE SUBSECTIONS REPORT

If the Autocreate Subsections Report was selected, the user is presented with the following prompt when the Report Output Setup Window closes:

```
Select Desired Autocreate Subsections Option:
    - Autocreate Subsections for all Sections (default)
    - Autocreate Subsections for selected Sections
    - Duplicate ReportDocument and Sections
```

Select the desired Autocreate option. With each option, a selection window opens to display the names of the *ReportDocuments* that are available for the report to process. Select the *ReportDocument* for which you want to autocreate sections

If you selected the option to autocreate subsections for selected sections only, another window will open to allow you to select the desired *Sections*. Select one or more *Sections* from the list and ***Accept*** the window. (Because the names of all *Sections* will be displayed, it is useful to set a filter to display only those with the desired suffix.) Any selected *Sections* that are not **contained by** the *ReportDocument* will be ignored.

At this point, the report begins creating subsections. It returns no messages to the user unless the report attempts to execute an invalid or defective *ReportBlock*. During execution, the report records the following information in the output file:

- ❑ Name of the *ReportDocument*
- ❑ Name of each *Section* that is a target of the **contains** relationship and the following attribute values: Number, Title, Print Report Block, and Autocreated/Output Subsections Blocks.
- ❑ List of *ReportBlocks* referenced
- ❑ List of errors detected

After report execution, examine the output file for any error messages. For example, the names of *ReportBlocks* that could not be found will be listed at the end of the file in an error message. The presence of this message indicates that the *Section* elements created by the Autocreate Subsections Report are not complete.

EXECUTING THE OUTPUT SECTIONS REPORT

If the Output Sections Report was selected, a selection window opens to display the names of the *ReportDocuments* that are available for the report to process. Select the *ReportDocument* that you want to output.

The report then prompts you to change the default settings for certain features. Select `Yes` or `No` at this prompt:

```
Do you want to Change any Run-Time Options?
- Yes
- No
```

If you select `Yes`, you will be presented with a series of prompts that will allow you to specify the following print parameters. If you click in an RDD-100 window while a prompt is displayed, the default value will be used and the next prompt displayed.

- ❑ Page Numbering (body)
    - ■ Sequential (default)
    - ■ By section, with a section number prefix

- ❑ Page Numbering (appendices)[1]
    - Sequential
    - By appendix, with an appendix letter prefix (default)
- ❑ Figure Numbering
    - Sequential (default)
    - By Sections
- ❑ Table Numbering
    - Sequential (default)
    - By Sections
- ❑ Section Text Display Format
    - Text within a section begins as a new “paragraph” (default)
    - Text within a section begins as a new “sentence”
- ❑ Graphical Diagrams to be printed in Capability Sections[2]
    - Behavior Diagrams
    - Functional Flow Block Diagrams (FFBDs)
    - Data Flows Diagrams (IDEF0s)
    - FFBD and IDEF0 Diagrams.
    - none (default)
- ❑ Reference Previous Paragraph Rather Than Duplicate?
    - No (default)
    - Yes

[1] This prompt is presented only if the default choice (`Number Pages Sequentially)` for the `Specify Page Numbering Method` prompt is selected.

[2] If a B1, B2, or SRS is to include control flow diagrams, the user must specify a graphical diagram output form. Also, the response to this prompt is overridden on output by the ***Suppress Diagrams*** option in the Report Output Setup window.

❑ Text Styling for Element Descriptions

- Document Default Styling (default)
- Styling “As Entered” within an Elements Description

EXECUTING REPORTS USING THE COMMAND READER

If you always choose the same report options when executing a report for a particular document, use the RDD-100 Command Reader to save steps. With the Command Reader, you can compose a command file that specifies the Design Guide Report to run, the *ReportDocument* to run it for, and the options that you normally enter at the report prompts.

To execute the command file, select the option ***Utilities > run command file*** from the RDD-100 Main Menu. This option opens a file list window labeled ***batch execution***. Here, you select the command file that you want to execute.

Figure 3-1 shows a sample command file that executes the Output Sections Report.

```
* COMMAND FILE FOR OUTPUT SECTIONS REPORT

set report option: action to: file

import data from file: SupportInfo/DesignGuide-TypeRReports/DGR-Reports.rpt
ownership: acquire history:as is

set report option: format to: postscript

answer report prompt for: element with: ReportDocument System/Segment Specification (SSS)

*   Do you want to change any Run-Time Options?
*   'No (Default)' 'Yes'

answer report prompt for: menu with: 'Yes'

*   Specify Page Numbering Method
*   'Number Pages Sequentially (default)' 'Number Pages by Sections'

answer report prompt for: menu with: 'Number Pages by Sections'

*   Specify Figure Numbering Method
*   'Number Figures Sequentially (default)' 'Number Figures by Sections'

answer report prompt for: menu with: 'Number Figures by Sections'

*   Specify Table Numbering Method
*   'Number Tables Sequentially (default)'
*   'Number Tables by Sections'

answer report prompt for: menu with: 'Number Tables by Sections'

*   Specify Section Text Display Format
*   'Text within a section begins as a new "paragraph" (default)'
*   'Text within a section begins as a new "sentence"'

answer report prompt for: menu with: 'Text within a section begins as a new "sentence"'

*   Specify Graphical Diagrams to be Printed in Capability Sections
*   'Behavior Diagrams' 'Functional Flow Block Diagrams (FFBD)'
*   'Data Flow Diagrams (IDEF0)' 'FFBD and IDEF0 Diagrams'
*   'none (default)'

answer report prompt for: menu with: 'Functional Flow Block Diagrams (FFBD)'

*   Reference Previous Paragraph Rather Than Duplicate?
*   'No (Default)' 'Yes'

answer report prompt for: menu with: 'Yes'

*   Specify Text Styling for Element Descriptions
*   'Document Default Styling (default)'
*   'Styling "As Entered" within an Elements Description'

answer report prompt for: menu with: 'Document Default Styling (default)'

run internal report: 'Output Sections' file: OutputSections.ps
```

FIGURE 3-1 Sample Command File for Running the Output Sections Repot

CREATING MULTIPLE DOCUMENTS OF THE SAME TYPE

A typical System Description Database may contain several *ReportDocument* elements of the same Document Type. For example, an SSS may be required both for the system and for each segment. You can use the Autocreate Subsection Report to duplicate the outline of an existing document; the new *ReportDocument* and new *Sections* are assigned all of the attributes and relationships of the original elements. The user can then tailor these characteristics to the needs of the new document.

To create a new document of the same type as another, select the Autocreate Subsection Report as described earlier in this chapter. This prompt appears:

```
Select Desired Autocreate Subsections Option

    - Autocreate Subsections for all Sections (default)
    - Autocreate Subsections for selected Sections
    - Duplicate ReportDocument and Sections
```

Select `Duplicate ReportDocument and Sections.`

When a selection window appears, select the *ReportDocument* to be processed by the Autocreate Subsections Report.

Now this prompt appears:

```
    Enter an Abbreviation for the ReportDocument to be Created.
```

In response, enter the *ReportDocument* Abbreviation to be used in generating the new *ReportDocument* and its associated *Sections*. This acronym should be unique. For an SSS or SSDD, this might be the acronym for the system or segment. If you press ***return*** or ***accept*** before entering an acronym, the original *ReportDocument* Abbreviation is used, and the newly-created *ReportDocument* and *Sections* have the identical names as the original elements with a suffix of .1 appended.

After you specify an abbreviation, this prompt appears:

```
        Do you want to duplicate the <Autocreated> Subsections ?
            Yes
            No (default)
```

Select “`Yes`” or “`No`”. If you select “`Yes`”, all *Sections* are duplicated. If you select “`No`”, the *Sections* that have an Author attribute value of `<Autocreated>` will not be duplicated. This is the normal way of creating duplicate document outlines.

Since the attributes and relationships from the original template are duplicated, modify the new template to link it to the appropriate System Engineering elements for the new document. For example, if only the non-autocreated *Sections* are duplicated, the user would normally adjust the *ReportDocument* attributes and the *ReportDocument* **reports on** and the *Sections* **uses** relationships. The user would then run the Autocreate Subsections Report to complete the new document outline.

EXECUTING THE PROJECT UNIQUE IDS REPORT

Like the other Design Guide Reports, the Project Unique IDs Numbering Report may be executed either as an internal or external report. The report initiation process is the same.

1. From the RDD-100 Main Menu, select ***Print>internal report*** or ***Print>external report***, then select the Project Unique IDs Report. This opens the Report Output Setup Window. Set the parameters as desired (normally the output file should be in ASCII format for ease of review) and click on ***File***.
2. A list of the facilities in your image now appears. From this list, select the facility that contains the elements for which numbers are to be assigned. If no single facility contains all the engineering elements, you must repeat the process for each facility that you need.
3. After you select a facility, RDD-100 displays a prompt for you to select either to re-number all elements or to add numbers only to those elements that do not already have them. Select the option that you want.
4. Next, you are prompted to specify a prefix to be placed on the identifier that is created (this may be left blank).

➤ **NOTE:** If your report includes a traceability matrix, you should limit the prefix to four characters. Longer prefixes may overflow from the matrix's RQMT ID column into adjacent columns. If this happens, either renumber the elements with a shorter prefix or modify the Output Sections Report to make the traceability matrix's RQMT ID column wider.

5. If you selected the re-number option in step 3, you will be prompted to specify a starting number. You must specify a value in response to this prompt (no default), and the number must be equal to or greater than 1.

The report then creates Project Unique identifiers. If any errors occur, they are written to the file specified in the Report Output Setup Window (see step 1). When the report completes execution, review the contents of that file if its length is other than 0 bytes.

Chapter

4

Use of DMF by Design Guide Reports

This chapter describes the roles of the DMF element types with respect to the Autocreate Subsections Report and the Output Sections Report. The descriptions are in alphabetical order by element type. For each element type, a brief definition is provided, followed by descriptions first of its attributes and then of its relationships.

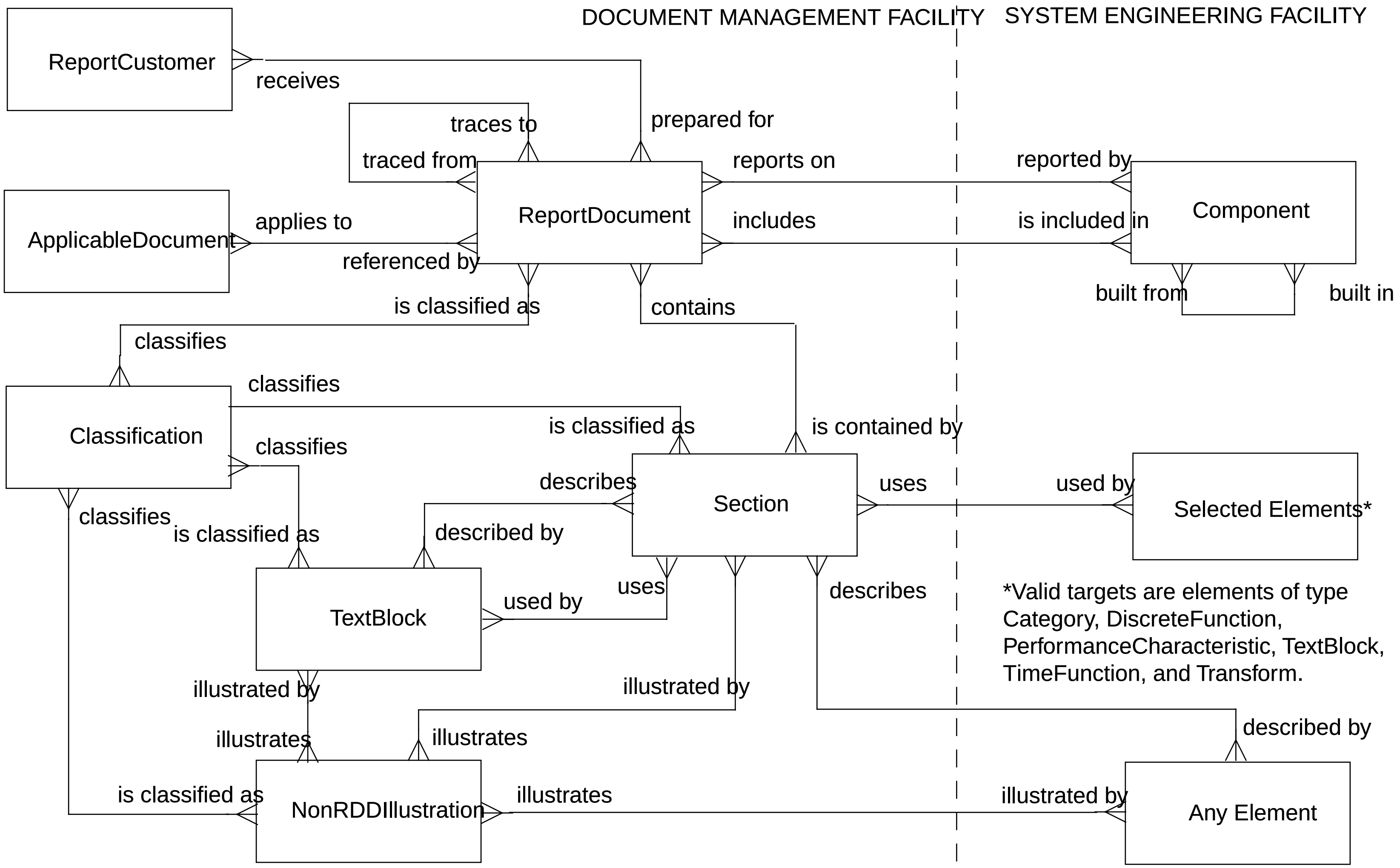
DMF ELEMENTS AND THEIR RELATIONSHIPS

The Document Management Facility contains the elements necessary to define the document structure (outline) for a particular specification. It forms the bridge between the engineering data that supports the system design and the Design Guide reports.

The DMF consists of seven element types:

- ❑ *ApplicableDocument*
- ❑ *Classification*
- ❑ *NonRDDIllustration*
- ❑ *ReportCustomer*
- ❑ *ReportDocument*
- ❑ *Section*
- ❑ *TextBlock*

Figure 4-1 illustrates the relationships between the DMF element types and other element types:



*Valid targets are elements of type Category, DiscreteFunction, PerformanceCharacteristic, TextBlock, TimeFunction, and Transform.

FIGURE 4-1 DMF Element Types and Relationships

APPLICABLEDOCUMENT

Identifies a document to be included in Section 2 of a specification. The element name and the Document Number attribute are printed.

ATTRIBUTES

Document Number

The reference number for the document.

Government

The category of the source of the document. The Output Sections Report uses the value of this attribute to determine whether to list the document in Section 2.1 or 2.2 of the *ReportDocument(s)* that is the target of the **applies to** relationship.

The value is selected from an enumerated list with the following selection options: `Federal`, `Military`, `Other Government Agency`, `Government`, and `Non-Government`. A value of other than `Non-Government` will cause the document to be listed in Section 2.1. With a value of `Non-Government`, the document will be listed in Section 2.2. If different categories are needed, corresponding changes must be made to both the enumerated list and to the *ReportBlocks*.

Publication Type

Type of publication. The Output Sections Report uses the value of this attribute to determine the heading under which the document will be listed in Section 2.1 or 2.2.

The value is selected from an enumerated list with the following selection options: `SPECIFICATION`, `STANDARD`, `DRAWING`, `Manual`, `Regulation`, `Handbook`, `Bulletin`, and `Other`. If different headings are required, corresponding changes must be made to both the selection values and to the *ReportBlocks*.

RELATIONSHIPS

applies to *ReportDocument(s)*

The *ReportDocument(s)* that references this *ApplicableDocument.*

CLASSIFICATION

Classification elements allow the user to designate classification markings to be printed on pages, section titles, table and figure titles, and paragraphs. If no *Classification* elements are provided for a document, the document is printed without classification markings.

The Abbreviation and Title attributes define the text to be used for the classification markings. The Usage attribute defines where classification markings are placed.

When a classification marking is to be applied to different parts of a document (for example, sometimes as a page marking, sometimes as a heading marking, and sometimes as a content marking) a different *Classification* element should be created for each usage of a particular classification. These different elements will have the same values for the Abbreviation and Title attributes but different values for the Usage attribute. For user convenience, the names of the elements should reflect the differences. For example, it may be necessary to have three different elements for an unclassified marking, named, for example, “Unclassified (title)”, “Unclassified (content)”, and “Unclassified (page)”. The Usage attribute of each would be “Title”, “Content”, and “Page” respectively. All three would have an Abbreviation of “(U)” and a Title of “Unclassified”.

ATTRIBUTES

Abbreviation
Character string to be used as a classification marking for section headings and contents (for example, “(U)” or “(S)”). The string should include the opening and closing parentheses. (See the Title attribute for the marking for pages.)

Rank
An integer value that indicates the relative level of the classification. The higher the number, the higher the classification level. Rank is used to determine the highest level classification within a document, on a page, or within a paragraph.

Title
The classification marking to be printed (centered) at the top and bottom of document pages. If blank, the name of the *Classification* element is used. (See the Abbreviation attribute for the marking for headings and contents.)

If the title is too long, it may overprint other information at the top and bottom of the page.

Usage
In conjunction with the targets of the **classifies** relationship, defines the scope of the classification marking. It is an enumerated list with the following selection values: `Title`, `Content`, `Page` and `nil`. See Table 4-1.

RELATIONSHIPS

classifies
NonRDD-Illustration, ReportDocument, Section, TextBlock, or *engineering elements*
Designates that some portion of the target element information is to be printed with classification markings.

TABLE 4-1 Classification Usage

TARGET ELEMENT TYPE FOR THE **classifies** RELATIONSHIP	VALUE OF *Classification* Usage ATTRIBUTE		
	TITLE	CONTENT	PAGE
Section	Print the Abbreviation attribute of the *Classification* after the section heading. Overrides a `Content` classification for the *ReportDocument*.	Print the Abbreviation attribute of the *Classification* after the section heading and before the section contents. Applies only to the contents of the *Section* Description attribute, not to targets of the **describes** relationship. Overrides a `Content` classification for the *ReportDocument*.	For level 1 *Sections* only, print the Title attribute of the *Classification* at the top and bottom of each page of the section. Overrides a `Page` classification for the *ReportDocument*.
engineering element or *TextBlock*	Used only by Autocreate Subsections Report; when Title attribute is copied to *Section* Title attribute, **is classified as** target is also linked if available.	Print the Abbreviation attribute of the *Classification* before the element description. Overrides a `Content` classification for the *ReportDocument*.	N/A
NonRDDIllustration	Print the Abbreviation attribute of the *Classification* after the figure or table title.	Although not printed, identifies the *Classification* of the content of the *NonRDDIllustration*.[1]	N/A
ReportDocument	Print the Abbreviation attribute of the *Classification* on the title page after the document title.	Print the Abbreviation attribute of the *Classification* after all section headings and before each paragraph. May be overridden on an individual element basis. Print the Abbreviation attribute of the *Classification* after all table and figure titles. Print the Title attribute of the *Classification* above all tables.	Print the Title attribute of the *Classification* on all pages in the document. May be overridden on any level 1 *Section*. [2]

[1] Should only be used when using page high markings for PostScript output.

[2] Page-high and document-high page marking are supported for PostScript output, but only document-high marking is supported for Interleaf output. (Interleaf does not support different headers and footers within a section.) To generate document-high markings, a *Classification* that **classifies** the *ReportDocument* must have its Usage attribute set to "`Page`". This *Classification* element is used to mark the top and bottom of each page of the document, regardless of the page content. In both PostScript and Interleaf output, this marking can be overridden for a level 1 Section (such as an Appendix). If no *Classification* element whose Usage attribute is set to "`Page`" **classifies** the *ReportDocument*, the document is printed with page-high markings. The Output Sections Report will determine the highest marking of the contents of the page and use this *Classification* to mark the top and bottom of the page.

NONRDDILLUSTRATION

NonRDDIllustration elements are used to designate space in the specification for figures and tables which have been generated outside of RDD-100. For each *NonRDDIllustration*, the Output Sections Report will either place the illustration in the report output or leave space in the output file for the illustration depending on the *NonRDDIllustration* attribute values. In either case, the Output Sections Report enters the illustration title and insertion point page in the List of Figures or List of Tables, as appropriate.

When incorporating an illustration in the output file, the illustration may be only a single-page graphic when the report is producing PostScript output. An Interleaf output file may incorporate single page graphics and multi-page documents. *NonRDDIllustrations* can only appear at the end of a *Section*.

ATTRIBUTES

Document Type

Format of the graphic or document to be incorporated. This is an enumerated attribute with the following selection options: `Postscript`, `Interleaf`, and `Complete Interleaf Document`. The option chosen must match the format of the document being generated. Otherwise, the *NonRDDIllustration* is ignored.

EPS Picture Height

Height of the graphic in decimal inches (PostScript output only). If this attribute or the EPS Picture Height are not set, the illustration will automatically be scaled to 7.8 inches high by 6.5 inches wide.

EPS Picture Width

Width of the graphic in decimal inches (PostScript output only). If this attribute or the EPS Picture Height are not set, the illustration will automatically be scaled to 7.8 inches high by 6.5 inches wide.

EPS Picture Scale in Percent

The scaling, in percent, to be applied to the graph (PostScript output only). The number must be a decimal and defaults to 100.0. This scaling is further factored to make certain that the illustration fits within the margins of the document.

Illustration Type

Type of illustration. This attribute is an enumerated list with the following selection values: `Figure`, `Table`, and `nil`. A value of `nil` is treated the same as a value of `Figure`.

This value controls the labeling of the illustration title and, for PostScript, the number sequence to be used.

Include File

Name of the file containing the single page graphic or multiple page document (Interleaf only) to be included in the report output file. If the file is not in the current directory, the full path name of the file must be supplied. If this attribute contains an entry, the Number of Pages attribute is ignored.

Number of Pages

Number of pages to be allowed for the insertion of the illustration if it is not being incorporated directly by the report (i.e., the Include File attribute is empty). This attribute must be an integer and defaults to 1.

Title

Name of the illustration to be included in the List of Figures or List of Tables, as appropriate. The title is printed on the output page as a place holder when the attribute Include File is empty. If this attribute is empty, the report will use the name of the element.

RELATIONSHIPS

illustrates *Section*, *TextBlock* or *engineering element*

If linked to a *Section*, space for the illustration is provided when generating the output based on the section information.

If linked to a *TextBlock* or engineering element, any *Section* which **describes** this element includes space for the illustration.

is classified as *Classification(s)*

Identifies a security classification to be used to either mark the illustration title or identify the classification of the content of the illustration depending on the *Classification* Usage attribute. This relationship must be established by the user, and the corresponding *Classification* Usage attribute set to produce the classification markings in the document. For details, see the description of the *Classification* element type.

REPORTCUSTOMER

The *ReportCustomer* element type is used to define information about the recipient of the document for printing on the document title page. The name of the *ReportCustomer* element and the values of the Department Code and Address attributes are printed beneath the “Prepared for:” line of the title page. If a *ReportCustomer* element is not linked to the *ReportDocument*, the “Prepared for:” text is omitted from the title page.

ATTRIBUTES

Address — Customer's address. Printed beneath the value in the Department Code.

Department Code — Customer's department code. Printed beneath the name of the ReportCustomer element.

RELATIONSHIPS

receives *Report-Document(s)* — The document(s) being prepared for the customer.

REPORTDOCUMENT

The *ReportDocument* element defines a particular document to be printed. Each Document Sections Template provided contains one *ReportDocument* element.

ATTRIBUTES

Abbreviation

The acronym for the document, such as SSS, SEG1, etc. The Autocreate Subsections Report uses this Abbreviation to ensure that the subsections it creates have a unique name that is linked to the associated *ReportDocument*. Each autocreated subsection has this acronym appended to the end of its name preceded by a period and followed by the symbol “@”.

In the Document Sections Template, this attribute is preset to match the acronym used as a suffix in the name of all the *Sections* that are targets of the **contains** relationship. This attribute should be customized by the user for the particular document to be produced.

When running the Output Sections Report, this value should always match the acronym in the suffix of all *Sections* **contained by** the *ReportDocument*.

CDRL Number

CDRL number to be printed on the title page of the document following the phrase “CDRL SEQUENCE NO.”. If this attribute is not set, the text “CDRL SEQUENCE NO.” will be omitted from the title page.

Contract Number

Number of the contract for which the report is being written. This is printed on the title page of the document following the phrase “CONTRACT NO.”. If this attribute is empty the text “CONTRACT NO.” will be omitted from the title page.

Contractor Address

Address of the contractor to be printed on the title page of the document beneath the Contractor Name.

Contractor Name

Name of the contractor to be printed beneath the “Prepared by:” line of the title page of the document. If this attribute is empty, both the “Prepared by:” phrase and the Contractor Address will be omitted from the title page.

Document Control Number

Reference number to be printed on the title page of the document and in the upper right hand corner of each page in the document.

Document Type

The type of specification being generated. This attribute is required and must have the value SSS, B1, B2, SSDD, IRS, or SRS, depending on the specification. This attribute has been pre-set in each Document Sections Template. This setting should not be changed by the user unless the user is also modifying the Design Guide reports; they test this attribute to control certain report actions. If Document Type is not defined, both the Autocreate Subsections and Output Sections reports print an error message and terminate.

Format Standard

The standard being followed in preparing the document. This attribute is an enumerated list with the following selection values: `nil`, `490A/2167A`, and `498`. Note: The *ReportBlocks* which are sensitive to this attribute interpret `nil` as `490A/2167A`.

Title

A phrase, such as “System Specification” or “Prime Item Development Specification” that identifies the specification type. It is printed as part of the title on the document title page. When the Output Sections Report prints the document, this phrase is automatically appended with the Title (or name) of the *Component* that is the target of the **reports on** relationship, and, if appropriate, with the Title or name of its ancestor whose Component Type = `System`, with suitable phrasing to form the complete document title.

This attribute has been preset in the Document Sections Templates. It needs to be changed only for an SSS or SSDD that documents a segment or subsystem; in that case the word Segment or Subsystem should be substituted for the word System which has been preloaded in the Document Sections Templates for each SSS and SSDD.

RELATIONSHIPS

contains
Sections

Identifies all the *Sections* that will either be processed by the Autocreate Subsections Report or printed in the document, including appendices. The reports only process subsections for *Sections* that are targets of the **contains** relationship.

Specifications may be generated at any time during the development cycle. This means that there will be times when specifications are generated even though some *Sections* will have no content. There is no harm in printing reports with sections for which no data exists. For them, *Section* headings will print, but with no content below the heading. If you do not want those sections to print, they may be unlinked as targets of the **contains** relationship. Later, as more data is added to the database, unlinked *Sections* can be re-linked to the *ReportDocument* to generate a more complete specification.

includes
Components

The *Components* to which requirements are allocated in an SSS, SSDD, or B1 specification. If no target is specified for this relationship, the Autocreate Subsections Report identifies the next lower-level *Components* for these specifications by using the **built from** relationship of the *Component* that is the target of the **reports on** relationship. Therefore, the **includes** relationship is a default override and should be used only if some *Component(s)* other than the next lower-level *Components* are to be documented in a report. Only one element in each branch of the component hierarchy should be identified (thus, do not specify as targets all descendants of the *Component* that is the target of the **reports on** relationship). If neither the **includes** nor the **built from** relationship have been established, the allocation portion of these documents is empty.

For an IRS, if targets are specified for the **includes** relationship, only the interfaces between the **reports on** and **includes** *Components* will be printed. If no targets are specified, the IRS will document all interfaces for the targets of the **reports on** relationship.

is classified as
Classification(s)

Identifies a security classification to be assigned to the *ReportDocument* Title or to used as the default for all section headings, sections contents, and document pages, depending on the value in the Usage attribute of the *Classification*. See the Usage Table in the description of the *Classification* element type for more information.

prepared for
ReportCustomer

The customer for whom the document is being produced. The *ReportCustomer* information is printed on the title page of the document.

referenced by
Applicable-Documents

The applicable documents to be listed in Section 2 of the document.

reports on
Component

The *Component* for which an SSS, SSDD, B1, B2, or SRS is to be generated. The **reports on** relationship is required. If this relationship is not established, both the Autocreate Subsections and Output Sections reports print an error message and terminate.

For an IRS, there may be more than one target for the **reports on** relationship. The IRS will document interfaces for all targets that are specified.

traced from
ReportDocument

The *ReportDocument(s)* from which requirements in the current document are to be traced when generating a Requirements Traceability Matrix. If absent, the matrix is generated assuming that the parent requirements are **documented by** a *Source*.

traces to
ReportDocument

Inverse of the relationship **traced from** *ReportDocument*.

SECTION

Section elements identify specific sections to be printed in a document. *Sections* may be created by importing a predefined Document Sections Template, by manually creating or augmenting a template, and/or by running the Autocreate Subsections Report against a partial template. These elements also store the names of the *ReportBlocks* that are used by the Autocreate Subsections Report and the Output Sections Report when they are acting on the *Section*.

The Document Sections Templates contain a *Section* element for each invariant section to be printed in the corresponding 490A/2167A or 498 document. In those templates, the name of a *Section* element has the form

title .specid

where *title* is the same as the Title attribute and *specid* is an acronym identifying the type of specification (such as “SSS”). A space is interposed between *title* and the period. For example, the *Section* element for section 1.2 of the SSS has the character string “System Overview” as its Title attribute, and its Number attribute is 1.2. Its name “System Overview .SSS”.

When the Autocreate Subsections Report creates a new *Section*, its name is set to the Title or, if the Title is empty, the name of the engineering element for which the *Section* is being created. This name is appended with a space followed by a period followed by the *ReportDocument* Abbreviation followed by the character “@”. If the resulting name is not unique an additional suffix is added. The additional suffix consists of a period followed by a unique integer.

In some cases, a standard *Section* is autocreated which is not associated with a particular engineering element (for example, section 3.X.2 of an IRS). In these cases, the name is set by the Autocreate *ReportBlock* to a standard pre-defined name appended with the appropriate suffix as described above.

ATTRIBUTES

Author

The Autocreate Subsections Report sets the Author attribute to “`<Autocreated>`” for all autocreated *Sections*.

Autocreate/ Output Subsections Blocks

The Autocreate/Output Subsections Blocks attribute is used by the Autocreate Subsections Report. It supplies that report with two pieces of information: the name of the *ReportBlock* that it executes to create subsections for this *Section*; and the name of an Output *ReportBlock* that will be assigned to the subsections it creates. This attribute may be blank, or it may contain the names of either one or two *ReportBlocks*.

- If it is blank, the *ReportBlock* named “`>@ generic`” is used, which will generate subsections when appropriate.

- If the first line contains the name of a *ReportBlock*, that name must identify the Autocreate *ReportBlock* that is to be used to create subsections. Autocreate *ReportBlocks* are identified by the first two characters of their names being “`>@`”. For example, “`>@ External Interfaces`” is the name of a valid Autocreate *ReportBlock*.

- If the second line contains the name of a *ReportBlock*, it must specify an Output *ReportBlock*. The Autocreate Subsections Report will copy the name of the Output *ReportBlock* to the Report Block Name attribute of its subsections. Output *ReportBlocks* are identified by the initial characters of their names being “`>Print`”. For example, “`>Print External Interfaces`” is the name of a valid Output *ReportBlock*. The names of the two Autocreate and Output *ReportBlocks* must be separated by a carriage return.

If lower level *Sections* are created manually, the user must copy this value into the Report Block Name attribute of any subsections that should be printed using this block.

The *ReportBlocks* referenced in this attribute in the *Sections* provided in the templates are merely suggested for use. The user may change these references to other blocks.

Description

This attribute can be used to store text that is not part of the engineering data but is desired in the specification (for example, an introduction or overview of a section). The Output *ReportBlocks* print the contents of the *Section* Description attribute as the first paragraph of the section.

Number

Hierarchical number for the *Section*. This number is printed as part of the section heading of the output and is entered into the document’s Table of Contents. The *Section* elements in the Document Sections Templates provided have been assigned Numbers that are consistent with the standard numbering of the specifications.

This value forms the basis for the numbering of all subsections and must be defined if the Autocreate Subsections Report is to be executed. In autocreated *Sections*, this attribute value is set to the next sequential hierarchical number based on the parent *Section’s* Number. If an Autocreate *ReportBlock* traverses a hierarchy of elements, creating subsections for several levels of elements, the created *Sections* form a hierarchy by number beneath the parent *Section*.

If the Number is not defined, the Autocreate Subsections Report assigns the value “`No.Number.Assigned`” to the *Section’s* Number and proceeds to the next *Section* without further processing. No subsections are created.

Report Block Name

Name of the *ReportBlock* to be used by the Output Sections Report to determine the printed contents of this section of the document. If this attribute is empty, the Output Sections Report uses the *ReportBlock* named “`>Print Description`” to print the section. The format or content of the data printed for this *Section* may be changed by specifying a different *ReportBlock*. The Output *ReportBlocks* are described in Appendix C, ”Output Sections Report Blocks”.

Normally, a value is assigned to this attribute by the Autocreate Subsections Report based on values in the Autocreate/Output Subsection Blocks of the parent *Section*.

The user may also assign a value to this attribute by typing in a *ReportBlock* name.

Title

Character string that is printed as the heading for the section and in the Table of Contents. If this attribute is empty, the Output Sections Report prints the name of the *Section* element as the section heading and enters that name in the Table of Contents.

In an Autocreated *Section*, the Title attribute is set to be the same as the Title attribute of the engineering element for which the *Section* is being created. If the Title attribute of that engineering element is empty, the *Section* Title is set to the name of the element. In some cases, a standard *Section* is autocreated which is not associated with a particular engineering element, (for example, section 3.X.2 of an IRS). In these cases, the Title is set by the Autocreate *ReportBlock* to a standard pre-defined name.

RELATIONSHIPS

describes *engineering element(s)* or *TextBlock(s)*

Element(s) to be used by the Output Sections Report to generate the contents of the section.

Unless otherwise stated in the description of the Autocreate *ReportBlock* (see Appendix B), the Autocreate Subsections Report establishes this relationship to the engineering element for which the *Section* is being created. This is established primarily when the default print logic is to be used or when a key element is needed to control the Output *ReportBlock* logic.

illustrated by *NonRDD-Illustration(s)*

An illustration not created by RDD-100 which is to be included in the output file or for which space needs to be provided in the printed output file.

is classified as *Classification*

Identifies a security classification to be used to mark the section heading, the section heading and contents, or all pages in a level 1 section, depending on the *Classification* Usage attribute. This relationship must be established by the user, and the corresponding *Classification* Usage attribute set to produce the classification markings in the document. See the Usage Table in the description of the *Classification* element type for more information.

is contained by *ReportDocument*

Identifies the *ReportDocument(s)* with which the *Section* is associated. Even if a *Section* **is contained by** more than one *ReportDocument*, its autocreated subsections will be linked automatically to only one *ReportDocument*.

uses *selected element(s)*

Specifies the starting element for creation of subsections by the Autocreate Subsections Report. For example, the *ReportBlock* named “`>@ generic`” builds subsections from engineering elements that are targets of the **uses** relationship. The following element types are valid targets: *Category*, *DiscreteFunction*, *PerformanceCharacteristic*, *Scenario*, *TextBlock*, *TimeFunction*, and *Transform*. Many Autocreate *ReportBlocks* allow multiple targets of the **uses** relationship.

Note that some *ReportBlocks* ignore targets of **uses** and start at a pre-determined element in the engineering data. For example, the *ReportBlock* named “`>@ SSDD HWCI Identification`” begins at *Components* with Component Type “`HWCI`” that are in the **built from** hierarchy of the *Component* **reported by** the *ReportDocument*.

If the required starting element is not specified or non-existent, new subsections will not be created when the parent Section is processed. Appendix B contains descriptions of the elements and relationships needed by each Autocreate *ReportBlock*.

TEXTBLOCK

The element type *TextBlock* is primarily used to store arbitrary text that is to be printed in a specification as separate, unnumbered paragraphs with differing classifications. The need for differing classifications makes usage of this element type necessary, since this cannot easily be entered and maintained in the *Section* Description attribute.

If, however, the user chooses to present this information as separate, numbered subsections, the corresponding *Sections* may be autocreated. To do so, the *TextBlocks* must be created and the following attribute and relationship established, as appropriate. In addition, for the parent *Section*, the Autocreate/Output Subsections Blocks attribute must be set to “`>@ TextBlocks`”.

ATTRIBUTES

Description

The text to be printed as a separate paragraph in a document section based on the *TextBlock* **described by** *Section* relationship.

Title

Value to be assigned to the Title of an associated autocreated *Section* (that is, the Autocreate Subsections Report assigns the value of this attribute to the Title attribute of the autocreated *Section*). If the Title attribute is blank, the *TextBlock* name is used.

The string is also used to create the new *Section* element Name. The string is first appended with a period followed by the *ReportDocument* Abbreviation, followed by the symbol @. If this Name is not unique in the database, sequences of .1 will continue to be appended until the Name is unique.

RELATIONSHIPS

described by *Section(s)*

Sections in which the *TextBlock* Description attribute is to be printed.

Using the default print logic (that is., no Report Block Name was entered for the *Section*), when multiple *TextBlocks* are described by a *Section*, each *TextBlock* Description is printed as a separate, unnumbered paragraph. The *TextBlocks* are sorted by Number and then by Name to order the paragraphs in the output. These paragraphs will follow the printing of the *Section* Description attribute and precede the printing of any other *Section* **describes** element information.

illustrated by *NonRDD-Illustration(s)*

An illustration stored externally to RDD-100 which is to be included in the report output file or for which space needs to be provided in the output file.

is classified as *Classification*

The *Classification* to be used when printing the Description attribute of the *TextBlock*. The *Classification* Usage attribute must have the value “`Content`”. See the explanation of *Classification* for further information.

used by *Section(s)*

The parent *Section(s)* for which subsections are to be created.

Chapter

5

Use of Engineering Elements by Design Guide Reports

For the Design Guide reports to properly process the SDD to produce a document, the system model in the RDD-100 database must have certain characteristics consistent with the Autocreate and Output *ReportBlocks* referenced in the Document Sections Templates provided with the reports. The engineering element types and their characteristics which are used by the pre-defined *ReportBlocks* are described in detail in this chapter. It includes the following sections:

INTRODUCTION

Using the Design Guide reports, a document is produced based on a document definition established using element types in the DMF. To successfully produce a document, DMF elements must be properly linked to elements in the System Description Database (SDD) and correctly reference *ReportBlocks* which are part of the Design Guide reports. To aid the user in this process, predefined templates have been provided which partially populate the DMF for each of the 490A/2167A and 498 documents supported by the Design Guide Type R reports. Appendix A documents the pre-defined templates and their references to *ReportBlocks*.

This chapter defines the element types, attributes, and relationships which need to be populated in the SDD to produce the complete content of the documents using the pre-defined Document Sections Templates and their referenced *ReportBlocks*. This chapter does not document the entire ISE schema -- just the element types, attributes, and relationships which are required or supported by the Autocreate and Output *ReportBlocks* associated with the Design Guide Type R Reports for the pre-defined templates. To support variations, additional Autocreate and Output *ReportBlocks* have been provided beyond those referenced in the pre-defined templates. Appendices B and C describe how each *ReportBlock* uses the ISE element types, relationships, and attributes.

OVERVIEW

In general, the default 490A/2167A and 498 Document Sections Template use the same SDD elements. The elements and primary relationships needed in the SDD to produce documents using the Design Guide Type R reports are illustrated in Figure 5-1 and detailed in the following sections. The element types *FNet*, *INet*, *RNet*, *TNet*, *Transform*, and *Condition* are recognized by the *ReportBlocks*; however, they are not called out separately in the following sections because the *ReportBlocks* use only the base set of element attributes and relationships for these element types and the relationships created automatically by building decompositions.

The following paragraphs provide general guidelines on how to populate the SDD in order to use the default Document Sections Templates. Further details are provided in the individual Autocreate and Output *ReportBlock* descriptions.

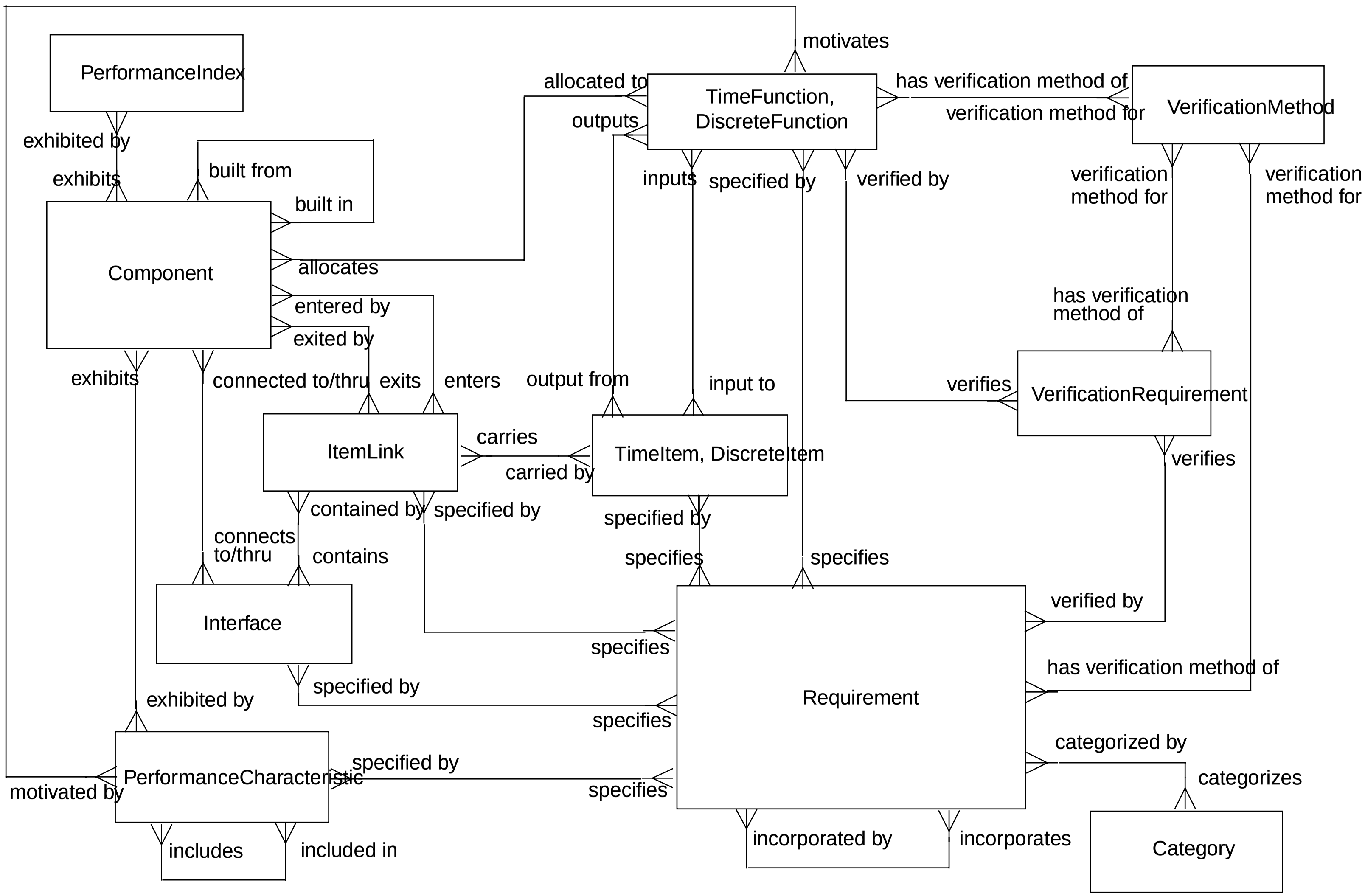


FIGURE 5-1 Primary Engineering Elements and Relationships

CAPABILITIES/FUNCTIONAL REQUIREMENTS

In order to use the Design Guide reports, functional requirements should be established by defining *Functions*. A functional requirement statement may be housed either in the Description of a *Function* or in a **specified by** *Requirement*. Individual functional performance requirements are stated in *Requirements* that **specify** a *Function*.

CAPABILITIES

The Design Guide reports support multiple interpretations of capability. A capability may be a single *Function* and its **specified by** *Requirements* or a collection of *Functions* and related *Requirements*.

In the former case, Autocreate *ReportBlocks* have been provided that use the *Function* hierarchy to determine what capability sections are to be generated.

In either case, *PerformanceCharacteristics* may be used to identify a capability as a collection of one or more *Functions* and their **specified by** *Requirements*. By their choice of Autocreate and Output *ReportBlocks*, users have a choice as to whether or not the individual *Functions* are identified in the specification. This approach provides a great deal of flexibility for the system level specification, particularly where requirements are state/mode dependent. Also, the Autocreate *ReportBlocks* for *PerformanceCharacteristics* allow a *PerformanceCharacteristic* to identify end-to-end requirements (that is, path timing requirements which have not been budgeted to individual *Functions*) against a *Component* for inclusion in the capabilities section of a specification. When using the *Function* only approach, such capability sections must be manually created. In addition, although not the preferred approach, a specification can be produced prior to modeling by linking *Requirements* directly to *PerformanceCharacteristics*.

The preferred approach, as reflected by the pre-defined Document Sections Templates, is to use *PerformanceCharacteristics* at the system level and *Functions* at the lower levels of specification where the visibility of data flow is important.

STATES AND MODES

PerformanceCharacteristics are used to identify system states and modes. For a 490A System specification, which is organized by states and modes, the *PerformanceCharacteristics* hierarchy must have as its leaf elements *PerformanceCharacteristics* of Type `Capability`. For those specifications which only require state/mode/capability tables, such as SRSs, the leaf *PerformanceCharacteristics* which link to *Functions* are not required to be of Type `Capability`.

TRACEABILITY

Document traceability between *Requirements* is established using the *Requirement* **incorporated by** *Requirement* relationship. When shall statements appear in *Function* Descriptions, the **justifies** relationship should be used to trace from a higher level *Requirement* to the *Function* hierarchy.

NON-FUNCTIONAL REQUIREMENTS

All non-functional requirements are established using *Requirements* elements that **specify** the subject *Component(s)*.

CATEGORIES

From a specification point of view, *Categories* are used to group *Requirements* for autocreation of sections. The Autocreate *ReportBlocks* support traversal of either the *Requirements* hierarchy or a *Category* hierarchy. In the first case, sections are created for *Requirements*; in the second case, the user controls whether or not to create sections for each *Requirement* by his choice of Autocreate *ReportBlock*. When separate *Requirement* sections are not created, there are several print format options for organizing multiple *Requirements*.

TRACEABILITY

Document traceability between *Requirements* is established using the *Requirement* **incorporated by** *Requirement* relationship.

INTERFACES

The Design Guide reports assume that the engineering elements pertaining to interfaces have been populated using the rules identified below. Although many Document Sections Templates assume that all external interfaces have been documented in IRSs, IDDs, or ICDs, an entire set of Autocreate *ReportBlocks* have been provided to autocreate sections through each level of interface definition: *Interfaces*, *ItemLinks*, and *Items*. In addition, these blocks and the variety of Output *ReportBlocks* should be sufficient to produce an ICD for which a pre-defined template is not provided.

EXTERNALINTERFACES

Starting at the system level, an *Interface* is defined for each external system with which the system interfaces and the relationship *Interface* **connects to** *Component* is established for the system and the external.

When the system *Component* is decomposed into **built from** *Components*, for each system-level external *Interface*, the **connects to** system target is replaced with the single *Component* which supports the *Interface*. At each level of the component hierarchy, this process is repeated. If there is no single *Component* which provides the logical connection, the *Interface* **connects to** relationship (to the system or a descendent *Component*) is left unchanged and *ItemLinks* are created for the *Interface* linking them to the pertinent *Components* with the **input to** or **output from** relationships, as appropriate, and to the *Interface* using the **is contained by** relationship. In a like manner, as the *Components* are further decomposed, relink the *ItemLinks* to lower level *Components*. There is normally no need to decompose the external system; however, if this is done, the same rules are applied to this portion of the component hierarchy.

Note: For modeling or specification purposes, it may be necessary to introduce *ItemLinks* at a higher level of component decomposition than described above. In this case, as the component hierarchy is further developed, both the *Interface* and *ItemLink* terminus points are adjusted downward in the hierarchy.

INTERNALINTERFACES

Internal interfaces are treated exactly like external interfaces from the perspective of the hierarchy level in which they are first identified.

ALLOCATION

In the Design Guide reports, allocation is reflected in allocation tables. The relationships used to control the contents of these tables is different for capabilities/functional requirements and non-functional requirements. When traversing *Function* and *Requirements* hierarchies (as opposed to *PerformanceCharacteristic* and *Category* hierarchies) to autocreate sections, these same relationships are used to control the stopping point for section creation.

CAPABILITIES/FUNCTIONAL REQUIREMENTS

Functional requirements are allocated to *Components* by allocating *Functions* (that is, *Function* **allocated to** *Component*). By definition, allocation of a *Function* to a *Component* also allocates any *Requirement* that **specifies** the *Function* regardless of how, if at all, the *Function* and *Requirement* are linked to *PerformanceCharacteristics*.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are allocated to *Components* by establishing the relationship *Requirement* **specifies** *Component*.

BASE SET OF ELEMENT CHARACTERISTICS

All engineering elements used to produce a document using the Design Guide Reports should have the following attributes and relationships defined, as appropriate. More details concerning the relationships to the DMF are discussed in Appendix B, "Autocreate Subsections Report Blocks", and Appendix C, "Output Sections Report Blocks".

ATTRIBUTES

Description — The description, definition, or requirements statement for the element, as appropriate. Text for a section is drawn from the Description attribute.

Title — The title to be used for the element in all documentation. If Title is empty, the element name is used.

Project Unique ID — The unique element identifier. If non-empty, this is printed in braces after printing the element Description in a paragraph and after printing the element name in a table. Note: The contents of many tables, such as the Requirements Traceability Table, are dependent on the Project Unique ID value.

RELATIONSHIPS

illustrated by *NonRDD-Illustration(s)* — Illustration stored externally to RDD-100 which is to be included in the output file or for which space needs to be provided in the printed output file whenever a *Section* which **describes** the element is printed.

is classified as *Classification* — The *Classification* to be used when printing the element Description attribute. The *Classification* Usage attribute must have the value: `Content`.

described by *Section(s)* — *Section(s)* for which either the element Description is printed or the element is used as a starting point for determining the information to print during execution of the Output Sections Report. Either the Output *ReportBlock* to be executed is designated by the *Section* Report Block Name attribute or, if no *ReportBlock* is specified, the default *ReportBlock* `>Print Description` is executed against the element.

used by *Section(s)* — *Section(s)* which will use the element as a starting point for autocreation of new *Sections* to form numbered subsections of the parent section. Either the Autocreate *ReportBlock* to be executed is designated by the *Section* Autocreate/Output Sections Blocks attribute or, if no *ReportBlock* is specified, the default *ReportBlock* `>@ generic` is executed against the element.

CATEGORY

From a specification point of view, *Categories* are used to group *Requirements* for auto-creation of *Sections*. In the Category .rdt files provided for ISE, there is a *Category* element for each pre-defined *Section* in the corresponding Document Sections Template file. Each *Category* element has the same name, number, and title as the corresponding *Section* element. In order to use these *Categories* in a Template, they must first be linked to the corresponding *Section*.

The Categories should have the following relationships for use with the Design Guide reports:

RELATIONSHIPS

categorizes *Requirements*
The *Requirements* that are of this type.

incorporated by *Category*
Indicates a hierarchical relationship between *Categories*. The scope of the subject (that is, lower level) *Category* is included in the scope of the object (higher level) *Category.*

incorporates *Category*
Inverse of **incorporated by** *Category.*

COMPONENT - TOP LEVEL

All documents produced by the Design Guide reports document a *Component* or, in the case of an IRS, possibly a set of *Components*. A *Component,* of Component Type `System`, is used to identify the system for which the SDD is being built. *Components* of other types are used to identify lower level constituents, such as subsystems, from which the system is to be built.

To produce a particular document using the Design Guide reports, the SDD must have a *Component* element, as appropriate for the document, with the following attribute values and relationships:

ATTRIBUTES

Abbreviation

The abbreviation for the system or component.

The Report Writer variable `TheReportedOnElementAbbreviation` is set to this value by the Output Sections Report. This variable may be referenced in the Description attribute of any element by enclosing the variable name in backslashes (specifically, `\TheReportedOnElementAbbreviation\`). When the Description is printed, the reference is replaced in the printed text with the value of the variable. An example of this would be to have the 1.1 Identification *Section* of all SRS *ReportDocuments* reference the same *TextBlock* which contains standard CSCI identification material, including the variable reference to print the particular CSCI abbreviation.

Component Type

The specific type of the *Component*.

The only values of Component Type for which the Design Guide Type R *ReportBlocks* explicitly check are `System`, `Environment`, `CSCI`, `FWCI`, `HWCI`, and `Human`.

Title

The documentation title to be used for the system or component. If Title is empty, the *Component* name is used.

The Report Writer variable `TheReportedOnElementName` will be set to the Title value by the Output Sections Report if the value is not empty; otherwise, this variable is set to the element name. This variable may be referenced (i.e., the name enclosed in back slashes) in the Description attribute of any element by enclosing the variable name in backslashes (i.e., `\TheReportedOnElementName\`). When the Description is printed, the reference is replaced in the printed text with the value of the variable. An example of this would be to have the 1.1 Identification *Section* of all SRS *ReportDocuments* reference the same *TextBlock* which contains standard CSCI identification material including the variable reference to print the particular CSCI title/name.

RELATIONSHIPS

built from *Components*

The immediate next level of components which constitute the subject component or system. You should have a hierarchy of *Component* targets of the **built from** relationship extending down from your top-level *Component (*for example, the segments of the System, the subcomponents of the segment, and so forth).

connected thru *Interfaces*

The logical interfaces between the top-level *Component* and external systems. This is a computed relationship that is established when the relationship *Component* **connected to** *Interface* is established from either the top-level component or a subcomponent in the component hierarchy below the top-level component (See **built from**).

has context *FNet*

The *FNet* which shows how the system or component interacts functionally with any external systems. To create a context *FNet*, you must create a *TimeFunction,* representing the behavior of the system and its externals, and build a behavior diagram consisting of a "root" *TimeFunction* for the system and each external, and of the top level *TimeItems* representing item flows between the root *Functions*. Then, the subject *Component* is linked to the *FNet* using the **has context** relationship.

entered by *ItemLink(s)*

The *ItemLink(s)* from which the *Component* obtains *Item(s).* This is a computed relationship that is established when the relationship *Component* **inputs from** *Interface* is established from either the top-level component or a subcomponent in the component hierarchy below the top-level component.

exhibits *Performance-Characteristics(s)*

The states (or modes) in which the component operates.

exited by *ItemLink(s)*

The *ItemLink(s)* to which the *Component* provides *Item(s)*. This is a computed relationship that is established when the relationship *Component* **outputs to** *Interface* is established from either the top-level component or a subcomponent in the component hierarchy below the top-level component.

performs *TimeFunction*

The *Component* "root" function which represents all functionality for the system or component. There should be at most one root function for a *Component*.

reported by *ReportDocument(s)*

The *ReportDocument(s)* that defines the document(s) to be produced for the *Component*.

COMPONENT - LOWER LEVEL

To produce a complete 490A/2167A SSS, B1, or SSDD or a complete 498 SSDD, you must identify the subcomponents (that is, subordinate elements) of the *Component* for which you are producing the document. (A subcomponent may be a top-level component for another document. See “Component - Top Level” on page 5-10 for the attributes and relationships which a component must have when used in this context.)

The subcomponents (*Components*) should have the following attributes and relationships:

ATTRIBUTES

Component Type
The specific type of the *Component*.

The only values of Component Type for which the Design Guide Type R *ReportBlocks* explicitly check are `System`, `Environment`, `CSCI`, `FWCI`, `HWCI`, and `Human`.

Percent of Memory Utilization
The total planned memory utilization of a *Component* which is a software or firmware processing resource.

Percent of Processor Utilization
The total planned processor utilization of a *Component* which is a software or firmware processing resource.

RELATIONSHIPS

allocates *TimeFunctions*/ *DiscreteFunctions*
The functional requirements to be achieved by this subcomponent. By definition, allocation of a *Function* to a *Component* also implicitly allocates any *Requirement* which **specifies** the *Function*.

built in *Component*
The next higher-level *Component* of which this *Component* is a partition or part.

built from *Components*
The next lower-level *Components* which comprise this *Component*.

connected thru *Interfaces*
The logical interfaces between the *Component* and other *Components*, which may be either internal or external to the system. This is a computed relationship that is established when the relationship *Component* **connected to** *Interface* is established from either this component or a subcomponent in the component hierarchy beneath this component.

entered by *ItemLink(s)*
The *ItemLink(s)* from which the *Component* obtains *Item(s).* This is a computed relationship that is established when the relationship *Component* **inputs from** *Interface* is established from either this component or a subcomponent in the component hierarchy beneath this component.

executes *Component(s)*
The *Component(s)* of Component Type `CSCI` or `FWCI` that executes on this *Component*, which should be of Component Type `HWCI`.

executed on *Component*
Inverse of **executes** *Component*.

exhibits *Performance-Index(s)*

Computer hardware characteristics of a processing resource (i.e., a *Component* of ComponentType `HWCI` which **executes** *Components* of ComponentType `CSCI` or `FWCI`).

exited by *ItemLink(s)*

The *ItemLink(s)* to which the *Component* provides *Item(s)*. This is a computed relationship that is established when the relationship *Component* **outputs to** *Interface* is established from either this component or a subcomponent in the component hierarchy beneath this component (See **built from**).

specified by *Requirement*

The non-functional characteristics required of the *Component*.

INTERFACE

Interfaces are used to define logical connections between the system and external systems, between subcomponents of the system, and between subcomponents of the system and of external systems.

Interfaces should have the following relationships:

connects to
Components
The elements which are logically connected by this *Interface.*

contains
ItemLinks
The *ItemLinks* which constitute the *Interface.*

specified by
Requirement
Required characteristics of the *Interface.*

ITEMLINK

ItemLinks define the logical or physical path that carries energy, matter, or information (*Items*) across a logical *Interface* between the system and external systems, between subcomponents of the system, and between subcomponents of the system and external systems.

ItemLinks should have the following relationships:

carries
TimeItems or
DiscreteItems
The *Items* which are passed across the link.

input to
Component
The component to which the *ItemLink* provides *Items*.

output from
Component
The component from which the *ItemLink* obtains *Items*.

is contained by
Interface
The *Interface* which the *ItemLink* supports.

specified by
Requirement
Required characteristics of the *ItemLink*.

PERFORMANCECHARACTERISTIC

PerformanceCharacteristics identify the states, modes, and capabilities of a component. Capabilities are the required functional actions of the system or component and the associated measurable performance parameters.

For 490A/2167A, a system's capabilities can be specified in the context of states in which the system can exist and the modes of operation within each state where requirements vary between states and/or modes.

For 498, a system may be described in terms of states only, modes only, states within modes, modes within states, or neither states nor modes.

ATTRIBUTE

Type

The specific type of the *PerformanceCharacteristic* selected from the enumeration list: `State`, `Mode`, and `Capability`.

RELATIONSHIPS

exhibited by
Component

The component for which this is a state (or mode, if states are not required). Only the top-level elements in the *PerformanceCharacteristics* hierarchy should be **exhibited by** the *Component*.

includes
Performance-Characteristic(s)

For a state, the modes of operation within that state or the capabilities that are to be available in the state.

For a mode, the capabilities that are to be available in this state or the states for the mode.

For a capability, any lower level constituent capabilities.

included in
Performance-Characteristic(s)

The inverse of **includes** *PerformanceCharacteristic*.

motivated by
TimeFunction/ DiscreteFunction

For a capability, the *Function(s)* and associated *Requirements* which constitute the capability.

specified by
Requirement(s)

The specific testable statements of required functional action and/or performance parameters which pertain to this capability (within its state and/ or mode, if defined). If a *PerformanceCharacteristic* is **motivated by** a *Function* that is **specified by** *Requirements*, it is assumed that all the *Requirements* pertain to the *PerformanceCharacteristic*. If only a subset of the *Requirements* pertain to the *PerformanceCharacteristic*, then this subset of *Requirements* should also **specify** the *PerformanceCharacteristic*.

PERFORMANCEINDEX

A *PerformanceIndex* identifies a processing resource characteristic such as memory size or processing speed.

A *PerformanceIndex* should have the following relationships:

exhibited by *Component* The component of which this is a characteristic.

REQUIREMENT

The *Requirement* elements contain the requirements statements for the system, both originating and derived. They contain the shall statements for the system and, as such, are the primary source for textual material in your 490A/2167A or 498 documents. In some cases, they are the formalization of requirements represented by other elements in the schema. Capabilities (i.e., functional requirements) for each *Component* are identified as collections of *TimeFunctions*/*DiscreteFunctions* and their **specified by** *Requirements*, where the *TimeFunctions/DiscreteFunctions* are **allocated to** lower-level *Components*.

Those functional *Requirements* allocatable to a *Component* are the sum of those which **specify** all of the *Functions* **allocated t**o the *Component*. A non-functional requirement or design constraint is identified as a *Requirement* which **specifies** *Component(s)*. An interface requirement is defined in terms of a physical *ItemLink* that **carries** *TimeItems* or *DiscreteItems* across a logical *Interface* which **connects to** the system and an external system or two components, where the *Interface/ItemLink* may be **specified by** *Requirements*.

The Design Guide Type R reports utilize the following attribute and relationships for *Requirements*:

ATTRIBUTE

Report Option

Controls how the *Requirement* Description is printed. The possible values are `begin Section`, `begin paragraph`, `begin lettered item`, and `follow previous text`.

- If the value is `begin Section` or `begin paragraph`, the *Requirement* Description will be printed as a new paragraph. If available the *Requirement* classification marking will be printed in front of the Description text and the *Requirement* Project Unique ID will be printed following the text and enclosed in square brackets.
- If the value is begin lettered item, the classification, Description, and Project Unique ID will be printed as a new lettered item within the current section.
- If the value is `follow previous text`, the Description followed by its Project Unique ID will be appended to the current text (i.e., either a paragraph or lettered item). The *Classification*, if available, for the *Requirement* will be compared to the current highest classification for any prior text, and the resultant highest ranked classification will take precedence for the combined text. The highest classification will be printed in front of the combined text. If no prior text exists, the *Requirement* information will be printed as the start of a new paragraph.

RELATIONSHIPS

categorized by
Category(s)

The type(s) of *Requirement* (i.e., `Originating`, `Non-Behavioral`, `Availability`, etc.).

documented by
Source

Identifies a source document for the *Requirement*. A *Source* must be linked only to *Requirements* which are not **incorporated by** other *Requirements* from the same source.

To be recognized as an originating source for traceability, the Source Type of the specified source document must have the value of either `Originating Requirements`, `Change Notice` or `Statement of Work.`

has verification method of
Verification-Method(s)

The method by which the component is to be tested to verify compliance with the requirement.

incorporates
Requirement(s)

Indicates a hierarchical relationship among *Requirements*. The scope of the subject (higher level) *Requirement* includes the scope of the object (lower level) *Requirement*.

incorporated by
Requirement

Inverse of **incorporates** *Requirement.*

justifies
TimeFunction/ DIscreteFunction

Identifies any *Function(s)* which was created based on the requirement. This establishes the traceability from an originating requirement to the *Function* hierarchy when shall statements are used in *Function* Descriptions.

specifies
Component, *DiscreteItem*, *Interface*, *ItemLink*, *Performance-Characteristic, TimeFunction/ DIscreteFunction/ Transform*

The *Component* to which a non-functional *Requirement* applies (that is, is allocated). Given the component hierarchy and a hierarchy of *Requirements*, the user must establish the mapping between the two. In the Design Guide reports, it is assumed that all *Requirements* linked to a *Category* used in a specification are allocated to the *Component* (**reported by** the *ReportDocument*) unless specifically allocated to a next lower level *Component(s)* by the **specifies** relationship. Therefore, explicit allocation is only done by exception to reflect that a specific *Requirement* (and its **incorporates** *Requirements*) in a hierarchy pertain only to a portion of the component hierarchy.

The *DiscreteItem*, *Interface*, *ItemLink*, *PerformanceCharacteristic, TimeFunction*, *DiscreteFunction* or *Transform* to which the specified *Requirement* pertains.

traces to
CriticalIssue(s), Decision(s)

An issue(s) or decision(s) arising from the requirement.

traced from
CriticalIssue(s), *Decision(s)*

A decision(s) or issue(s) which prompted the requirement statement.

verified by
Verification-Requirement

Any special test or examination which will be used to verify compliance with the *Requirement*.

SOURCE

Sources identify and house information about documents external to the SDD which have bearing on the system design or design process. Because *Sources* form part of the requirements traceability, the Design Guide Reports recognize these elements when printing traceability matrices.

Source elements are also used to generate document references to Interface Control Documents, Interface Requirements Specifications, and Interface Design Documents generated outside of RDD. The attributes and relationships which the Output Sections Report utilizes for a *Source* are:

ATTRIBUTES

Document Control Number

The reference number for the source document.

Source Type

The type of documentation.

Note: To be recognized as an originating source for traceability, the Source Type must be either `Originating Requirements`, `Statement of Work,` or `Change Notice`. To be recognized as an interface reference document, the Source Type must be either `ICD`, `IDD,` or `IRS`.

RELATIONSHIP

documents
Decision(s), Interface, Requirement(s)

The *Decision(s)* or *Requirement(s)* for which this source contains the originating information.

The *Interface* for which this source provides documentation.

TIMEFUNCTION AND DISCRETEFUNCTION

In addition to the relationships that result from building a system model using Behavior Diagrams, *TimeFunctions* and *DiscreteFunctions* should have the following relationships:

allocated to
Component

The *Component* which is to achieve the function and its requirements. By definition, allocation of a *Function* to a *Component* also allocates any *Requirement* which **specifies** the *Function*.

Only one *Function* on any branch of a component functional hierarchy should be allocated to an individual lower level *Component*.

has verification method of
Verification-Method(s)

The method by which the component is to be tested against the requirement, if any, stated in the *Function* Description.

inputs
TimeItems, *DiscreteItems*

Items that are input to the function.

justified by
Requirement

Establishes the traceability from a *Function* back to the originating *Requirement* which established the need for the *Function*. Only one *Function* in any branch of the function hierarchy should have this relationship.

motivates
Performance-Characteristic(s)

The capabilities of which this function is a part.

outputs
TimeItems, *DiscreteItems*

Items that the function produces.

performed by
Component

The component for which the function is the “root” function and therefore represents all functionality for the component.

specified by
Requirement(s)

The applicable parameter(s) associated with achieving the functional capability.

verified by
Verification-Requirement(s)

Any special test or examination of the component which is required to determine compliance with the requirement, if any, stated in the *Function* Description.

TIMEITEM, DISCRETEITEM, ABSTRACTOBJECTTYPE AND ATTRIBUTETYPE

In addition to the relationships that result from building a system model using Behavior Diagrams, appropriate elements should have the following attributes and relationships.

ATTRIBUTES

If *DiscreteItems*, *AbstractObjectTypes*, and *AttributeTypes* are used to do data modeling for an IRS and SRS, this additional information is required for *AttributeTypes* to reflect characteristics of the data contents:

Units
The units of measurement for this data.

Minimum Value
The minimum value that the data can assume.

Maximum Value
The maximum value that the data can assume.

Accuracy
The accuracy required for the data element, expressed as a percentage of the actual value

Precision
The precision or resolution of the element in terms of significant bits.

Domain
The type of data, selected from this enumeration list: `Boolean`, `Complex`, `Integer`, `Float`, `String`, and `Double`.

If *TimeItems* and *DiscreteItems* are used to do data modeling for an IRS and SRS, this additional information is required for *DiscreteItems* to reflect characteristics of the item contents:

Units
The units of measurement of the item information content.

Range
The range of values that the item can assume.

Accuracy
The accuracy required for the data element, expressed as a percentage of the actual value

Resolution
The precision or resolution of the element in terms of significant bits.

Data Type
The type of data content of the item selected from the enumeration list: `Boolean`, `Float`, `Integer`, `String`, `Text`, `IArray`, `FArray`, `SArray`, `Blob (undefined abstract)`, `File (ASCII, binary, etc.)`, `Record`, and `Database`.

If a *DiscreteItem* is specified as having structure, its Description attribute is used to summarize the structural content.

RELATIONSHIPS

carried by
ItemLink
The link that passes the *Item* across an interface.

specified by
Requirement
Required characteristics of the *DiscreteItem*.

VERIFICATIONMETHOD

A *VerificationMethod* identifies the general means used to determine that a component meets particular requirements and the level of testing at which the determination is made by that means. A separate *VerificationMethod* should be defined for each method type and verification level combination. *VerificationMethods* should have the following attributes and relationship established:

ATTRIBUTES

Method — The type of method selected from the enumeration list: `Inspection`, `Analysis`, `Demonstration`, `Test`, and `N/A`.

Verification Level — The phase of integration and test (i.e., `System`, `Segment`, `Prime Item`, `Configuration Item`, `Unit` or `Module`) at which the *VerificationMethod* is applied to determine that a component meets particular requirements.

RELATIONSHIP

verification method for *DiscreteFunction, Requirement, TimeFunction, Transform, Verification-Requirements* — The requirements to which this method and level applies.

VERIFICATIONREQUIREMENT

A *VerificationRequirement* is a special test or examination required to determine compliance of a component with its requirements.

A *VerificationRequirement* should have the following relationships:

has verification method of *VerificationMethod* — The method by which compliance is determined.

verifies *Discrete-Functions, Requirements, TimeFunctions, Transforms* — The requirements which are addressed.

Chapter

6

Customization

Users have a wide range of options for customizing the Design Guide reports and their utilization of the Document Management Facility (DMF). This chapter provides some guidelines for different forms of customization. It is assumed that the reader have a basic familiarity with the reports and the DMF. For more complex customization, the reader will need familiarity with the RDD-100 Report Writer.

This chapter includes the following sections:

MODIFYING DOCUMENT SECTIONS TEMPLATES

Each of the Document Sections Templates provided with the Design Guide Type R reports is a starting point for a particular document. It is a basic outline for the particular 4090A/2167A and 498 document which can be completed by manually creating subsections and/or by using the Autocreate Subsections Report. Any of these basic outlines may also be modified by:

- Changing the organization of the document by changing the section outline.
- Referencing different Autocreate *ReportBlocks*.
- Referencing different Output *ReportBlocks*.

In each case, one would follow the same general procedure:

1. Import a Document Sections Template which is close to the document outline that you want.
2. Delete, create, renumber, and/or rename *Sections* as needed to get the required outline.
3. Review each *Section* for the references to Autocreate and Output *ReportBlocks*. Adjust these as needed based on the *ReportBlock* descriptions in Appendix B and Appendix C.
4. If you want to save this customized template for reuse, save it as a separate .rdt file. To be comparable to the pre-defined Document Sections Templates, it should be saved without any links to engineering elements. For example, *ReportDocument* **reports on** and *Section* **describes** and **uses** relationships should not be present in the .rdt file.

If there is no Document Sections Template which suits your purpose, you can produce a new outline by creating the new *ReportDocument* and *Sections* and continue with step 3 above. Be sure to review the information in Chapter 4 to identify all the information needed in the *ReportDocument* and *Section* elements before running either the Autocreate Subsections or Output Sections report.

In modifying a template, one should be aware of the following:

- In some cases, implicit dependencies exist between the Autocreate *ReportBlocks* and the Output *ReportBlocks.* For example, if you have chosen an Autocreate *ReportBlock* which creates separate *Sections* for each *Function* associated with a *PerformanceCharacteristic*, you would not want to use an Output *ReportBlock* which prints *Function* Descriptions within a capability section. Doing so would result in each *Function* Description being printed once under the capability and again in a separate numbered paragraph.
- All of the Autocreate *ReportBlocks* derive the *Section* Numbers for autocreated *Sections* from the Number of the starting *Section*. Thus, they will correctly autocreate if they are used in a *Section* with a different number than those in the pre-defined Document Sections Templates.
- None of the Output *ReportBlocks* check for specific *Section* Numbers. Thus, they will print correctly if they are used in a *Section* with a different number than those in the pre-defined Document Sections Templates.

- The traceability list built by the Output *ReportBlocks* in order to generate a Traceability Matrix checks for specific *Section* Numbers based on the DocumentType and Format Standard. If you have changed your outline to include requirements in other *Sections* than those used in these standards, you will need to modify the select nodes which check the *Section* Number within the *ReportBlock* "`build TraceabilityList`".
- If you chose to create a new document type (for example, a new *ReportDocument* with DocumentType other than SSS, SSDD, B1, B2, IRS, or SRS), changes may be required to the reports.

MODIFYING THE DESIGN GUIDE REPORTS

There are two parts to the Design Guide reports:

- **Design Guide Reports Engine:** This is the application-independent portion of the reports. It processes the *ReportDocument* and *Sections* information in the DMF and invokes the Autocreate and Output *ReportBlocks* as directed by the DMF. The **Engine** is defined to include the two default *ReportBlocks* `>@ generic` and `>Print Description` and the special *ReportBlock* `>Print Previously Referenced Paragraph`.

- **Autocreate and Output ReportBlocks**: These are the application-dependent portions of the report (i.e., methodology, engineering schema, and document). These, respectively, contain the logic to create subsections and print sections according to predefined rules associated with the engineering schema and the standard being followed. Thus, there are differences between the Autocreate and Output *ReportBlocks* used in the 490A/2167A and 498 templates.

  The Autocreate and Output *ReportBlocks* are not in-line *ReportBlocks* within either the Autocreate Subsections Report or the Output Sections Report. They are invoked by using a *ReportIndirectBlock* in the body of the respective report. The specific Autocreate or Output *ReportBlock* which is invoked for any specific *Section* is controlled by the *Section's* attributes.

Customization of the reports, if needed, will most often involve changes to the Autocreate or Output *ReportBlocks* rather than to the **Engine**. The types of modifications may vary in complexity from changing the styling of paragraph titles to generating new Autocreate or Output *ReportBlocks*. However, the general process recommended when changing the reports would not vary. One should follow the steps listed below:

1. Import the Design Guide reports and any test data you may need into an image which contains the Design Guide Type R schema.
2. Erase history using ***Utilities>erase database history.*** Leave ***Maintain Database History*** turned on in ***Session>set user preferences***.
3. Modify the reports as needed and test the changes. It might be useful to keep notes on the changes you are making. If you are creating new Autocreate or Output *ReportBlock* logic, it is often useful to start with an existing blocks logic. Specifically, if you want to preserve the original logic (since pieces of it may be used elsewhere):
   - Create a new Autocreate or Output *ReportBlock*.
   - Copy the logic from an existing Autocreate or Output *ReportBlock* which is close to the logic you need. Walk through this logic exchanging *ReportDomainSets*, et al as needed. Unless you are absolutely certain that a lower level *ReportBlock* is not used elsewhere or that you will never need the original logic, do not modify an original *ReportBlock*. Simply follow these steps recursively, creating initial copies of the blocks and changing the copies as needed.
4. Delete any *DiagramDescriptions* and then export the modifications as an export to

superior. This will give you a good start on applying your changes to any later releases of the Design Guide Type R reports.

5. After exporting your changes, erase history, and save your new report using either ***File>save database*** or ***File>save elements*** or the `ReportHierarchy.rqt` report found in the <*RDD directory*>`/SupportInfo/General/Utilities` directory.

The following sections provide detailed information on characteristics of the Design Guide Type R reports which you may find useful in tailoring the *ReportBlocks*.

AUTOCREATE REPORTBLOCKS

The Autocreate *ReportBlocks* provided as part of the Design Guide Type R reports follow the conventions of the **Engine** to create new *Sections*. The differences between Autocreate *ReportBlocks* are in what *Sections* are created and how this is determined, not in how *Sections* are created. Specifically, they use the same primitives (i.e., specific Reporting Facility elements) to create new *Sections* and set their attributes and relationships. The user is referred to the *ReportBlock* `>@ generic` for the basic logic for creating a *Section*. These are the primitives which should be used in any new Autocreate *ReportBlocks.*

The following Report Writer variables are set by the **Engine** and may be useful in modifying/creating Autocreate *ReportBlocks:*

ThisDocument — Current *ReportDocument* which is being processed by the report.

TheReportedElement — Target of the *ReportDocument* **reports on** relationship.

BuildFromList — *Components* **built in** the *System*/*Component* which is **reported by** the *ReportDocument* -- unless this has been overridden by establishing the *ReportDocument* **includes** *Components*, in which case, contains the targets of the **includes** relationship.

OUTPUT REPORTBLOCKS

The Output *ReportBlocks* provided as part of the Design Guide Type R reports follow the conventions of the **Engine** for printing basic *Section* information. They use the same Reporting Facility elements to print standard section titles and element Descriptions that are used in the **Engine** default Output *ReportBlock* `>Print Description.` To produce a uniform document and utilize the **Engine** logic for such document wide features as classification markings, these same elements should be used in any new Autocreate *ReportBlocks.*

Description

For the element designated by the variable `CurrentElement`, prints the element's Description (if not empty) as a separate paragraph. If the paragraph is the first paragraph printed for the section, it may begin printing on the same line as the *Section* Title depending on the user selected run-time parameters. The printed Description is preceded by a classification marking, if available. The classification information maintained by the **Engine** to mark page headers and footers is updated. If the element is not a *Section* or *TextBlock*, the printed *Description* is followed by the element's Project Unique ID.

Print Section "describes" Description

For the current *Section*, prints the Description attributes (if not empty) of all non*TextBlocks* **described by** the *Section*. It implements the **Requirements Formatting Rules** for printing *Requirements* Descriptions.

Print Standard Section Titling

For the current *Section*, prints the *Section* Number and *Section* Title (or name) and the Description attributes (if not empty) of the following elements in the order listed: *Section* and all *TextBlocks* **described by** the *Section*.

The *Section* Title is preceded by the associated classification marking, if available. This *ReportBlock* also contains the logic which starts level 1 sections on a new page, resets headers and footers for classification changes, and inserts *NonRDDIllustrations.*

This block uses the *ReportBlock* `Description`, described above, to print the Description attributes.

The Report Writer variables described below are set by the **Engine** and may be useful in modifying/creating Output *ReportBlocks*.

➤ **NOTE:** Any of the variables which contain lists (i.e., multiple entries) are **OrderedCollections** and may be used in a *ReportDomainSet* of Type `Variable`. The user is referred to the *RDD-100 On-line Help* for details on building and accessing **OrderedCollections.**

BuildFromList

Components **built in** the *System*/*Component* which is **reported by** the *ReportDocument* -- unless this has been overridden by establishing the *ReportDocument* **includes** *Components*, in which case, contains the targets of the **includes** relationship.

CurrentSection

Current *Section* being processed.

ElementList

Targets of the current *Section's* **describes** relationship that are not *TextBlock* elements.

TheReportedElement

Target of *ReportDocument* **reports on** relationship.

ThisDocument

The *ReportDocument* being output.

TABLE CONVENTIONS

The Output *ReportBlocks* often use tables to present information. For uniformity, tables use the following conventions:

- All table numbers are printed from the variable **TableNumber**. Before printing the table, this variable is set by using the *ReportBlock* `create table number`.
- All table titles are printed from the variable **TableName**. This is set using a *ReportDefinition* in the Output *ReportBlock* before printing the table. The table name may be preceded by a classification marking contained in the variable **ClassificationFlag**. This variable is set before printing the table by using the common *ReportBlock* `check default classification for text and title`.
- The column headings (i.e., titles) are set before printing the table. They are set into the variables **Col1**, **Col2**, etc. To change the column headings, locate the single *ReportDefinition* preceding the table which sets the headings and edit this element.
- In most Output *ReportBlocks* using tables, the information to be printed in the table is collected before starting to print the table. The information will be placed in variables which are **OrderedCollections** and, in some cases, **OrderedCollections** of **OrderedCollections**.

MINIMIZING REPORT SIZE

In some environments, after customization of the Autocreate and Output *ReportBlocks*, there may be a need to reduce the size of the reports. The Design Guide reports are structured so that the Report Hierarchy query can be used to produce a subset of the reports. One can minimize the size by following these steps:

1. Save a copy of the entire report using ***File>export database*** or ***File>export elements***. If you decide later to make additional changes to the reports, you should use this master copy and then repeat the steps identified below to produce your working subset.
2. In the Autocreate Subsections Report, locate the *ReportBlock* `list of create blocks called indirectly`. If you open a Behavior Diagram of this block, you will see that it consists of a single select node where the false branch references all of the Autocreate *ReportBlocks* provided with the report. This branch is never executed. It is there to allow the Report Hierarchy query to locate all of the logic which, in totality, comprises the report. Thus, by modifying this branch, you can control what is exported as part of the report. Modify this branch to only reference those Autocreate *ReportBlocks* which you are using. Make the same type of modification to *ReportBlock* `list of output report blocks called indirectly` in the Output Sections Report.
3. Erase history.
4. Import the `ReportHierarchy.rqt` file found in the `<RDD directory>/SupportInfo/General/Utilities` directory.
5. Select ***File>export to subordinate***. Execute the Report Hierarchy query against the *ReportMainBlocks* `Autocreate Subsections` and `Output Sections`. Complete the export.
6. Test your report to make certain that you exported everything that is being used in your documents.
7. Since the Autocreate and Output *ReportBlocks* have many elements in common, the export to subordinate file will create ghosts to parts of the master report which were not exported. If you wish to remove these ghosts and further reduce the size of the report, take the following actions:
    a. Import the export to subordinate .rdt into a new image. Execute the Ghost Buster Report found in the `<RDD directory>/SupportInfo/General/Utilities/GhostBuster.rpt` file. This report should be executed using ***Print>external report*** and will remove all ghosts from your report.
    b. Erase history.
    c. Export the de-ghosted report using ***File>export database*** or ***File>export elements***. This is your minimized, customized report.

The process, as described above, produces a production subset of the entire report. You can use a similar process to tailor a subset for each document type you produce, such as an SSS, SSDD, etc. To do this, you would always start with your master report and, in step 2, change the name of the *ReportMainBlocks* to indicate the type of document and only reference the Autocreate and Output *ReportBlocks* used in that document type.

Appendix

A

Document Sections Templates

This documents the contents of the Design Guide Type R predefined templates for the 490A/2167A and 498 specifications. The chapter includes these sections

INTRODUCTION

Using the Design Guide reports, a document is produced based on a document definition established using element types in the DMF. To successfully produce a document, DMF elements must be properly linked to elements in the System Description Database (SDD) and correctly reference *ReportBlocks* which are part of the Design Guide reports. To aid the user in this process, predefined templates have been provided which partially populate the DMF for each of the 490A/2167A and 498 documents supported by the Design Guide Type R reports.

In this chapter, each predefined template is documented in the form of a tabular listing of the template *Sections* showing:

- ❑ The section number and title which is printed.
- ❑ Any pre-defined *ReportBlock* which is used to print the content of the section (that is, the value of the Report Block Name attribute).
- ❑ Any predefined *ReportBlock* used to autocreate subsections and its associated print *ReportBlock*, if any (that is, the value of the Autocreate/Output Sections Blocks attribute).

ELEMENT TYPE VS. DOCUMENT SECTIONS TEMPLATE

In general, the default 490A/2167A and 498 Document Sections Templates use the same SDD elements. Table A-1 shows which element types are used by the default 490A/2167A and 498 Document Sections Templates provided with the reports. The table is marked with an A and/or P to indicate which Design Guide report, either Autocreate Subsections or Output Sections, respectively, uses the element type. When an element type is marked with a P, this indicates that elements of this type are used in determining the information to print, but not necessarily that any information about the elements of this type are actually printed (i.e. an element of this type may be included in a chain of relationships which determine an element to be printed). The symbols (A) and (P) mean that use of the element type is supported but not required. The following sections address each of the Document Sections Templates.

TABLE A-1 Element Types for Design Guide Type R Documents

ELEMENT TYPE	490A / 2167A						498			
	SSS	B1	B2	SSDD	IRS	SRS	SSS	SSDD	IRS	SRS
Top-level *Component*	A/P	A/P	A/P	A/P	A/P	A/P	A/P	A/P	A/P	A
Peer *Component*	A	A	A	A/P	A/P	A	A	A	A/P	A
Sub*Component*	A/P	A/P		A/P				A/P		
PerformanceCharacteristic	A/P			(P)		(A/P)	A/P			A/P
TimeFunction, *DiscreteFunction*		A/P	A/P	P		A/P	(A/P)	P		A/P
Transform						(P)				(P)
FNet (structure graph)		A/(P)	A/(P)			A/(P)				A/(P)
FNet (context)	P	P	P	P						

TABLE A-1 Element Types for Design Guide Type R Documents (Continued)

ELEMENT TYPE	490A / 2167A						498			
	SSS	B1	B2	SSDD	IRS	SRS	SSS	SSDD	IRS	SRS
Requirement	A/P	A/P	A/P	P	A/P	AP	A/P	P	A/P	A/P
Category	A	A	A	A		A	A			A
Interface	A/P	A/P	A/P	A/P	A/P	A/P	A/P	A/P	A/P	A/P
ItemLink				P	A/P	A/P		A/P	A/P	A/P
TimeItem, *DiscreteItem*				P	A/P	A/P		A/P	A/P	A/P
AbstractObjectType, *AttributeType*					P	P			P	P
VerificationMethod	P	P	P			P	P			P
VerificationRequirement	A/P	A/P	A/P			A/P	A/P			A/P
PerformanceIndex				P				P		
Source	(P)	(P)	(P)	(P)		(P)	(P)	(P)		(P)

490A/2167A DOCUMENT SECTIONS TEMPLATES

SYSTEM/SEGMENT SPECIFICATION (SSS)

The Document Sections Template provided as a starting point for generating a 490A/ 2167A System/Segment Specification (SSS) is derived from DID number DI-CMAN-80008A and contains a *Section* for each invariant section of the document. The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report.

The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. In addition, the engineering elements must be linked in a manner consistent with the *ReportBlocks* which are used during section autocreation and printing. Illustrated below are the engineering element types and relationships which are used by the pre-defined *ReportBlocks* referenced in the 490A/2167A SSS Document Sections Template.

Table A-2 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name), and
- ❑ names of the *ReportBlocks* which are used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively.

TABLE A-2 490A/2167A System/Segment Specifiation (SSS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTIONS BLOCKS
1	Scope		
1.1	Identification		
1.2	System Overview		
1.3	Document Overview		
2	Applicable Documents		

TABLE A-2 490A/2167A System/Segment Specifiation (SSS)
Document SectionsTemplate (Continued)

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTIONS BLOCKS
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	System Requirements		
3.1	Definition	>Print Reported On Element’s Description & Context Diagram	
3.2	Characteristics		
3.2.1	Performance Characteristics		>@ PCs >Print PCs
3.2.2	System Capability Relationships	>Print State, Mode, Capability/ Fcn Table	
3.2.3	External Interface Requirements		>@ External Interfaces >Print Interfaces w Reference
3.2.4	Physical Characteristics		>@ Categories
3.2.5	System Quality Factors		
3.2.5.1	Reliability		>@ Categories
3.2.5.2	Maintainability		>@ Categories
3.2.5.3	Availability		>@ Categories
3.2.5.4	Additional Quality Factors		>@ Categories
3.2.6	Environmental Conditions		>@ Categories
3.2.7	Transportability		>@ Categories
3.2.8	Flexibility and Expansion		>@ Categories
3.2.9	Portability		>@ Categories
3.3	Design and Construction		
3.3.1	Materials		>@ Categories
3.3.2	Electromagnetic Radiation		>@ Categories
3.3.3	Nameplates and Product Marking		>@ Categories
3.3.4	Workmanship		>@ Categories
3.3.5	Interchangeability		>@ Categories
3.3.6	Safety		>@ Categories
3.3.7	Human Engineering		>@ Categories
3.3.8	Nuclear Control		>@ Categories
3.3.9	System Security		>@ Categories

TABLE A-2 490A/2167A System/Segment Specifiation (SSS) Document SectionsTemplate (Continued)

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTIONS BLOCKS
3.3.10	Government Furnished Property Usage		>@ Categories
3.3.11	Computer Resource Reserve Capacity		>@ Categories
3.4	Documentation		>@ Categories
3.5	Logistics		>@ Categories
3.6	Personnel and Training		
3.6.1	Personnel		>@ Categories
3.6.2	Training		>@ Categories
3.7	Characteristics of Subordinate Elements		>@ SSS Subordinate Elements >Print Subordinate Element Rqmts Table
3.8	Precedence		
3.9	Qualification		>@ Categories
3.10	Standard Sample		>@ Categories
3.11	Preproduction Sample, Periodic Production Sample, Pilot, or Pilot Lot		>@ Categories
4	Quality Assurance Provisions		
4.1	Responsibility for Inspection		
4.2	Special Tests and Examinations		>@ VRs >Print VRs
4.3	Requirements Cross Reference	>Print Requirements Verification Matrix	
5	Preparation for Delivery		>@ Categories
6	Notes		
6.1	Intended Use		
6.1.1	Missions		
6.1.2	Threat		
6.2	Acronyms and Abbreviations		
10	Appendix A - Requirements Traceability Matrix	>Print Requirements Traceability Matrix	

PRIME ITEM DEVELOPMENT SPECIFICATION (B1)

The Document Sections Template provided as a starting point for generating a Prime Item Development Specification (B1) is derived from MIL-STD-490A and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* which are used during section autocreation and printing.

Table A-3 lists the information content of the pre-defined *Sections* of this template, including the following:

- section number to be printed (i.e., *Section* Number)
- section title to be printed (i.e., *Section* Title)
- name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name)
- names of the *ReportBlocks* which are used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively.

TABLE A-3 490A2167A Prime Item Development Specification (B1) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
2	Applicable Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Requirements		
3.1	Prime Item Definition	>Print Reported On Element's Description & Context Diagram	
3.1.1	Prime Item Diagrams	>Print Reported On Element's Top-Level Behavior	>@ - Thru Allocated/Leaf Functions >Print Function, I/O, graph and graph references
3.1.2	Interface Definition		

TABLE A-3 490A2167A Prime Item Development Specification (B1) Document Sections Template (Continued)

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.1.2.1	External Interface Definition		>@ External Interfaces >Print Interfaces w Reference
3.1.2.2	Internal Interface Definition		>@ Internal Interfaces >Print Interfaces w Reference
3.1.3	Major Component List	>Print Major Component List	
3.1.4	Government Furnished Property List		
3.1.5	Government Loaned Property List		
3.2	Characteristics		
3.2.1	Performance		>@ - Only Allocated/Leaf Functions, Rqmts
3.2.2	Physical Characteristics		
3.2.2.1	Size		>@ Categories
3.2.2.2	Weight		>@ Categories
3.2.3	Reliability		>@ Categories
3.2.4	Maintainability		>@ Categories
3.2.5	Environmental Conditions		
3.2.5.1	Natural Environments		>@ Categories
3.2.5.2	Induced Environments		>@ Categories
3.2.6	Transportability		>@ Categories
3.3	Design and Construction		
3.3.1	Materials, Processes, and Parts		>@ Categories
3.3.2	Electromagnetic Radiation		>@ Categories
3.3.3	Nameplates and Product Marking		>@ Categories
3.3.4	Workmanship		>@ Categories
3.3.5	Interchangeability		>@ Categories
3.3.6	Safety		>@ Categories
3.3.7	Human Performance/Human Engineering		>@ Categories
3.4	Documentation		>@ Categories
3.5	Logistics		
3.5.1	Maintenance		>@ Categories

TABLE A-3 490A2167A Prime Item Development Specification (B1) Document SectionsTemplate (Continued)

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.5.2	Supply		>@ Categories
3.5.3	Facilities and Facility Equipment		>@ Categories
3.6	Personnel and Training		
3.6.1	Personnel		>@ Categories
3.6.2	Training		>@ Categories
3.7	Major Component Characteristics		>@ Major Component Characteristics[1]
3.8	Precedence		
4	Quality Assurance Provisions		
4.1	General		
4.1.1	Responsibility for Tests		
4.1.2	Special Tests and Examinations		>@ VRs >Print VRs
4.2	Quality Conformance Inspections		
4.2.1	Requirement Cross Reference	>Print Requirements VerificationMatrix	
5	Preparation for Delivery		>@ Categories
6	Notes		
10	Appendix A - Requirements Traceability Matrix	>Print Requirements Traceability Matrix	

[1] This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

3.7.x.1 `>Print Major Component Performance Characteristics`
3.7.x.2 `>Print Major Component Physical Characteristics`.

CRITICAL ITEM DEVELOPMENT SPECIFICATIONS (B2)

The Document Sections Template provided as a starting point for generating a Critical Item Development Specifications (B2) is derived from MIL-STD-490A and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* which are used during section autocreation and printing.

Table A-4 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name)
- ❑ names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively.

TABLE A-4 490A/2167A Critical Item Development Specification (B2) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
2	Applicable Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Requirements		
3.1	Critical Item Definition	>Print Reported On Element’s Description & Context Diagram	
3.1.1	Critical Item Diagrams	>Print Reported On Element's Top-Level Behavior	>@ - Thru Allocated/Leaf Functions >Print Function, I/O, graph and graph references
3.1.2	Interface Definition		>@ External Interfaces >Print Interfaces w Reference

TABLE A-4 490A/2167A Critical Item Development Specifcation (B2) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.1.3	Government Furnished Property List		
3.1.4	Government Loaned Property List		
3.2	Characteristics		
3.2.1	Performance		>@ - Only Allocated/Leaf Functions, Rqmts
3.2.2	Physical Characteristics		
3.2.2.1	Size		>@ Categories
3.2.2.2	Weight		>@ Categories
3.2.3	Reliability		>@ Categories
3.2.4	Maintainability		>@ Categories
3.2.5	Environmental Conditions		
3.2.5.1	Natural Environments		>@ Categories
3.2.5.2	Induced Environments		>@ Categories
3.2.6	Transportability		>@ Categories
3.3	Design and Construction		
3.3.1	Materials, Processes, and Parts		>@ Categories
3.3.2	Electromagnetic Radiation		>@ Categories
3.3.3	Nameplates and Product Marking		>@ Categories
3.3.4	Workmanship		>@ Categories
3.3.5	Interchangeability		>@ Categories
3.3.6	Safety		>@ Categories
3.3.7	Human Performance/Human Engineering		>@ Categories
3.4	Documentation		>@ Categories
3.5	Logistics		
3.5.1	Maintenance		>@ Categories
3.5.2	Supply		>@ Categories
3.5.3	Facilities and Facility Equipment		>@ Categories
3.6	Precedence		
4	Quality Assurance Provisions		

TABLE A-4 490A/2167A Critical Item Development Specifcation (B2) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
4.1	General		
4.1.1	Responsibility for Tests		
4.1.2	Special Tests and Examinations		>@ VRs >Print VRs
4.2	Quality Conformance Inspections		
4.2.1	Requirements Cross Reference	>Print Requirements Verification Matrix	
5	Preparation for Delivery		>@ Categories
6	Notes		
10	Appendix A - Requirements Traceability Matrix	>Print Requirements Traceability Matrix	

SYSTEM/SEGMENT DESIGN DOCUMENT (SSDD)

The Document Sections Template provided as a starting point for generating a System/Segment Design Document (SSDD) is derived from DID number DI-CMAN-80534 and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* which are used during section autocreation and printing.

Table A-5 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name)
- ❑ names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively.

TABLE A-5 490A2167A System/Segment Design Document (SSDD) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
1.1	Identification		
1.2	System Overview	>Print Reported On Element's Description	
1.3	Document Overview		
2	Referenced Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Operational Concepts		
3.1	Mission		
3.1.1	User Needs		
3.1.2	Primary Mission		

TABLE A-5 490A2167A System/Segment Design Document (SSDD) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.1.3	Secondary Mission		
3.2	Operational Environment		
3.3	Support Environment		
3.3.1	Support Concept		
3.3.2	Support Facilities		
3.3.3	Supply		
3.3.4	Government Agencies		
3.4	System Architecture		
3.4.1	System Elements	>Print Major Component List	
3.4.2	System External Interfaces	>Print Reported On Element's Context Diagram	>@ External Interfaces >Print SSDD System External Interfaces
3.4.3	System Internal Interfaces	>Print Reported On Element's Component View	>@ Internal Interfaces >Print SSDD System Internal Interfaces
3.5	Operational Scenarios	>Print State, Mode, CI Table	
4	System Design		
4.1	HWCI Identification		>@ SSDD HWCI Identification[1]
4.2	CSCI Identification		>@ SSDD CSCI Identification[1]
4.3	Manual Operations Identification		>@ SSDD Manual Operations Identification[1]
4.4	Internal Interfaces		
4.4.1	HWCI-to-HWCI Interfaces		>@ SSDD HWCI-to-HWCI Interfaces >Print SSDD Internal Interfaces
4.4.2	HWCI-to-CSCI Interfaces		>@ SSDD HWCI-to-CSCI Interfaces >Print SSDD Internal Interfaces
4.4.3	CSCI-to-CSCI Interfaces		>@ SSDD CSCI-to-CSCI Interfaces >Print SSDD Internal Interfaces
5	Processing Resources		>@ SSDD Processing Resources >Print SSDD CI Utilization Table
6	Quality Factor Compliance		
7	Requirements Traceability	>Print SSDD Requirements Traceability Matrix	
8	Notes		

1. This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

   4.x.y.1 `>Print SSDD CI Allocated Requirements`
   4.x.y.2 `>Print SSDD CI Design Constraints`
   4.x.y.3 `>Print SSDD CI System External Interfaces`

INTERFACE REQUIREMENTS SPECIFICATION (IRS)

The Document Sections Template provided as a starting point for generating a 490A/2167A type of Interface Requirements Specification (IRS) is derived from DID number DI-CMAN-80026A and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* used during section autocreation and printing.

Table A-6 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ the section number to be printed (i.e., *Section* Number),
- ❑ the section title to be printed (i.e., *Section* Title),
- ❑ the name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name), and
- ❑ the names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively

TABLE A-6 490A/2167A Interface Requirements Specification (IRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
1.1	Identification	>Print IRS Interface Identification	
1.2	System Overview		
1.3	Document Overview		
2	Applicable Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Interface Specification		>@ IRS Interfaces[1]
4	Quality Assurance		

TABLE A-6 490A/2167A Interface Requirements Specification (IRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
5	Preparation for Delivery		
6	Notes		

1. This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

3.1	`>Print IRS Interface Diagrams`
3.x	`>Print IRS Interface, Rqmts, ItemLink Table`
3.x.y.2	`>Print IRS ItemLink Message Table`
3.x.y.2.z	`>Print Item Definition, Rqmts`.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

The Document Sections Template provided as a starting point for generating a Software Requirements Specification (SRS) is derived from DID number DI-CMAN-80025A and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* used during section autocreation and printing.

Table A-7 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name)
- ❑ names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively.

TABLE A-7 490A/2167A Software Requirements Specifiation (SRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
1.1	Identification		
1.2	CSCI Overview	>Print Reported On Element's Description	
1.3	Document Overview		
2	Applicable Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Engineering Requirements		
3.1	CSCI External Interface Requirements		>@ External Interfaces, ILs

TABLE A-7 490A/2167A Software Requirements Specifiation (SRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.2	CSCI Capability Requirements	>Print State, Mode, Capability/ Fcn Table	>@ - Thru Leaf Functions >Print Function, Rqmts, I/O, graph and graph references
3.3	CSCI Internal Interfaces	>Print CSCI Internal Interfaces	
3.4	CSCI Data Element Requirements		
3.4.1	CSCI Internal Data Elements	>Print SRS Internal Data Table	>@ SRS Data Elements Internal >Print Item Definition
3.4.2	CSCI External Data Elements	>Print SRS External Data Table	>@ SRS Data Elements External >Print Item Definition
3.5	Adaptation Requirements		
3.5.1	Installation-Dependent Data		>@ Categories
3.5.2	Operational Parameters		>@ Categories
3.6	Sizing and Timing Requirements		>@ Categories
3.7	Safety Requirements		>@ Categories
3.8	Security Requirements		>@ Categories
3.9	Design Constraints		>@ Categories
3.10	Software Quality Factors		>@ Categories
3.11	Human Performance/Human Engineering Requirements		>@ Categories
3.12	Requirements Traceability	>Print Requirements Traceability Matrix	
4	Qualification Requirements		
4.1	Qualification Methods		
4.2	Special Qualification Requirements		>@ VRs >Print VRs
4.3	Requirements Cross Reference	>Print Requirements Verification Matrix	
5	Preparation for Delivery		>@ Categories
6	Notes		

498 DOCUMENT SECTIONS TEMPLATES

SYSTEM/SUBSYSTEM SPECIFICATION (SSS)

The Document Sections Template provided as a starting point for generating a 498 type of System/Subsystem Specification (SSS) is derived from DID number DI-IPSC-81431 and contains a *Section* for each invariant section of the document. The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report.

The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. In addition, the engineering elements must be linked in a manner consistent with the *ReportBlocks* which are used during section autocreation and printing.

Table A-8 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* Report-Block Name), and
- ❑ names of the *ReportBlocks* which are used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively.

TABLE A-8 498 System/Subsystem Specification (SSS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTIONS BLOCKS
1	Scope		
1.1	Identification		
1.2	System Overview	>Print Reported On Element's Description	
1.3	Document Overview		
2	Referenced Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	

TABLE A-8 498 System/Subsystem Specification (SSS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTIONS BLOCKS
3	Requirements		
3.1	Required States and Modes	>Print State, Mode, Capability/ Fcn Table	>@ PCs - States/Modes Only
3.2	System Capability Requirements		>@ PCs - Capabilities Only >Print PCs
3.3	System External Interface Requirements		>@ External Interfaces[1] >Print Interfaces w Reference
3.4	System Internal Interface Requirements		>@ Categories
3.5	System Internal Data Requirements		>@ Categories
3.6	Adaptation Requirements		>@ Categories
3.7	Safety Requirements		>@ Categories
3.8	Security and Privacy Requirements		>@ Categories
3.9	System Environmental Requirements		>@ Categories
3.10	Computer Resource Requirements		
3.10.1	Computer Hardware Requirements		>@ Categories
3.10.2	Computer Hardware Resource Utilization Requirements		>@ Categories
3.10.3	Computer Software Requirements		>@ Categories
3.10.4	Computer Communications Requirements		>@ Categories
3.11	System Quality Factors		
3.11.1	Reliability		>@ Categories
3.11.2	Maintainability		>@ Categories
3.11.3	Availability		>@ Categories
3.11.4	Portability		>@ Categories
3.11.5	Additional Quality Factors		>@ Categories
3.12	Design and Construction Constraints		
3.12.1	Physical Characteristics		>@ Categories

TABLE A-8 498 System/Subsystem Specification (SSS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTIONS BLOCKS
3.12.2	Transportability		>@ Categories
3.12.3	Flexibility and Expansion		>@ Categories
3.12.4	Materials		>@ Categories
3.12.5	Electromagnetic Radiation		>@ Categories
3.12.6	Nameplates and Product Marking		>@ Categories
3.12.7	Workmanship		>@ Categories
3.12.8	Interchangeability		>@ Categories
3.12.9	Nuclear Control		>@ Categories
3.12.10	Government Furnished Property Usage		>@ Categories
3.13	Personnel-related Requirements		
3.13.1	Human Engineering		>@ Categories
3.14	Training-related Requirements		>@ Categories
3.15	Logistics-related Requirements		>@ Categories
3.16	Other Requirements		
3.16.1	Documentation		>@ Categories
3.17	Packaging Requirements		>@ Categories
3.18	Precedence and Criticality of Requirements		
4	Qualification Provisions		
4.1	Special Tests and Examinations		>@ VRs >Print VRs
4.2	Requirements Cross Reference	>Print Requirements Verification Matrix	
5	Requirements Traceability	>Print Requirements Traceability Matrix	
6	Notes		

1. For a 498 template, this *ReportBlock* automatically creates an additional *Section* numbered Base.1, where Base is the Number of the parent *Section*, and assigns `>Print External Interface Identification` to the Report Block Name attribute of the *Section*.

SYSTEM/SUBSYSTEM DESIGN DESCRIPTION (SSDD)

The Document Sections Template provided as a starting point for generating a System/Subsystem Design Description (SSDD) is derived from DID number DI-IPSC-81432 and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* which are used during section autocreation and printing.

Table A-9 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* Report-Block Name)
- ❑ names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively

TABLE A-9 498 System/Subsystem Design Description (SSDD) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
1.1	Identification		
1.2	System Overview	>Print Reported On Element's Description	
1.3	Document Overview		
2	Referenced Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	System-wide Design Decisions		
4	System Architectural Design		
4.1	System Components	>Print Major Component List	

TABLE A-9 498 System/Subsystem Design Description (SSDD) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
4.1.1	HWCI Identification		>@ HWCI subordinate elements for SSDD[1]
4.1.2	CSCI Identification		>@ CSCI subordinate elements for SSDD[1]
4.1.3	Manual Operations Identification		>@ Human subordinate elements for SSDD[1]
4.1.4	Specification Tree		
4.2	Concept of Execution		
4.3	Interface Design		
4.3.1	External Interfaces		>@ External Interfaces[2] >Print Interfaces w Reference
4.3.2	Internal Interfaces		>@ Internal Interfaces, ILs, Items[3] >Print Item Definition
5	Requirements Traceability	>Print SSDD Requirements Traceability Matrix	
6	Notes		

1. This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

   4.1.x.y.1 `>Print SSDD CI Allocated Requirements`

   4.1.x.y.2 `>Print SSDD CI Design Constraints`

   4.1.x.y.3 `>Print SSDD CI Utilization Table`
2. For a 498 template, this *ReportBlock* automatically creates an additional *Section* numbered Base.1, where Base is the Number of the parent *Section*, and assigns `>Print External Interface Identification` to the Report Block Name attribute of the *Section*.
3. For a 498 template, this *ReportBlock* automatically creates an additional *Section* numbered Base.1, where Base is the Number of the parent *Section*, and assigns `>Print Internal Interface Identification` to the Report Block Name attribute of the *Section*.

INTERFACE REQUIREMENTS SPECIFICATION (IRS)

The Document Sections Template provided as a starting point for generating a 498 type of Interface Requirements Specification (IRS) is derived from DID number DI-IPSC-81434 and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent with the *ReportBlocks* used during section autocreation and printing.

Table A-10 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ the section number to be printed (i.e., *Section* Number),
- ❑ the section title to be printed (i.e., *Section* Title),
- ❑ the name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name), and
- ❑ the names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., Section Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively

TABLE A-10 498 Interface Requirements Specification (IRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
1.1	Identification	>Print IRS Interface Identification	
1.2	System Overview		
1.3	Document Overview		
2	Referenced Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Requirements		>@ IRS Interfaces[1]
4	Qualification Provisions	?	

TABLE A-10 498 Interface Requirements Specifcation (IRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
5	Requirements Traceability	?	
6	Notes		

1. This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

3.1	`>Print IRS Interface Diagrams`
3.x	`>Print Interface, Rqmts, ItemLink Table`
3.x.y.2	`>Print IRS ItemLink Message Table`
3.x.y.2.z	`>Print Item Definition, Rqmts`.

For a 498 template, the last 3.x *Section* created is not based on a *Interface* and, therefore, it's Report Block Name is unassigned.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

The Document Sections Template provided as a starting point for generating a498 type of Software Requirements Specification (SRS) is derived from DID number DI-IPSC-81433 and contains a *Section* for each invariant section of the document.

The user must complete the outline by adding *Sections* either manually or by using the Autocreate Subsections Report. The document *ReportDocument* and *Sections* should be linked to the appropriate engineering elements to complete the document definition. Further, the engineering elements need to be linked consistent the *ReportBlocks* used during section autocreation and printing.

Table A-11 lists the information content of the pre-defined *Sections* of this template, including the following:

- ❑ section number to be printed (i.e., *Section* Number)
- ❑ section title to be printed (i.e., *Section* Title)
- ❑ name of the *ReportBlock* used to print the section content (i.e., *Section* ReportBlock Name)
- ❑ names of the *ReportBlocks* used to autocreate subsections for this section and to print those subsections (i.e., *Section* Autocreate/Output Subsections Blocks).

If there is no *ReportBlock* listed, the default *ReportBlock* is used (i.e., `>@ generic` for autocreation and `>Print Description` for printing).

Detailed descriptions of each Autocreate and Output *ReportBlock,* including the element types and relationships used, are provided in Appendix B and Appendix C, respectively

TABLE A-11 498 Software Requirements Specification (SRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
1	Scope		
1.1	Identification		
1.2	System Overview		
1.3	Document Overview		
2	Referenced Documents		
2.1	Government Documents	>Print (2.1) Government Documents List	
2.2	Non-Government Documents	>Print (2.2) Non-Government Documents List	
3	Requirements		
3.1	Required States and Modes	>Print State, Mode, Capability/ Fcn Table	>@ PCs - States/Modes Only

TABLE A-11 498 Software Requirements Specification (SRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.2	CSCI Capability Requirements		>@ - Thru Leaf Functions >Print Function, Rqmts, I/O, graph and graph references
3.3	CSCI External Interface Requirements		>@ External Interfaces[1] >Print Interfaces w Reference
3.4	CSCI Internal Interface Requirements	>Print CSCI Internal Interfaces	
3.5	CSCI Internal Data Requirements	>Print SRS Internal Data Table	>@ SRS Data Elements Internal >Print Item Definition
3.6	Adaptation Requirements		
3.6.1	Installation-Dependent Data		>@ Categories
3.6.2	Operational Parameters		>@ Categories
3.7	Safety Requirements		>@ Categories
3.8	Security and Privacy Requirements		>@ Categories
3.9	CSCI Environment Requirements		>@ Categories
3.10	Computer Resource Requirements		
3.10.1	Computer Hardware Requirements		>@ Categories
3.10.2	Computer Hardware Resource Utilization Requirements		>@ Categories
3.10.3	Computer Software Requirements		>@ Categories
3.10.4	Computer Communications Requirements		>@ Categories
3.11	Software Quality Factors		>@ Categories
3.12	Design and Implementation Constraints		>@ Categories
3.13	Personnel-related Requirements		>@ Categories
3.14	Training-related Requirements		>@ Categories
3.15	Logistics-related Requirements		>@ Categories
3.16	Other Requirements		>@ Categories
3.17	Packaging Requirements		>@ Categories

TABLE A-11 498 Software Requirements Specification (SRS) Document SectionsTemplate

NUMBER	TITLE	REPORT BLOCK NAME	AUTOCREATE/OUTPUT SUBSECTION BLOCKS
3.18	Precedence and Criticality of Requirements		
4	Qualification Provisions		
4.1	Qualification Methods		
4.2	Special Qualification Methods		>@ VRs >Print VRs
4.3	Requirements Cross Reference	>Print Requirements Verification Matrix	
5	Requirements Traceability	>Print Requirements Traceability Matrix	
6	Notes		

1. For a 498 template, this *ReportBlock* automatically creates an additional *Section* numbered Base.1, where Base is the Number of the parent *Section*, and assigns `>Print External Interface Identification` to the Report Block Name attribute of the *Section*.

Appendix

B

Autocreate Subsections Report Blocks

This appendix describes the predefined Autocreate *ReportBlocks* provided with the Design Guide Type R reports. These *ReportBlocks* are used with a *ReportDocument*, a base set of parent *Sections* and the Autocreate Subsections report to create additional *Sections* of the document based on the engineering elements in the SDD.

Common attribute, relationship and target element type requirements are defined elsewhere in this document. The individual Autocreate *ReportBlock* definitions specify only attributes, relationships and target element types unique to that *ReportBlock*.

SPECIAL LISTS DEFINITIONS

BUILD FROM LIST

Each *ReportDocument* **reports on** a primary *Component*. A document may require an additional selection of *Components*, subordinate elements of the primary *Component*, to be identified. These are the *Components* to which requirements are allocated in the document. The collecting of these allocation *Components* is done only once in the Autocreate Subsections Report, prior to the processing of all individual *Sections*. In this appendix, this collection of *Components* will be referred to as the **Build From List**.

Two different relationship paths can be used to define the **Build From List**. The primary path is *ReportDocument* **reports on** *Component* **built from** the next level of *Component* elements. The **built from** relationship is NOT followed recursively. Only the first level of **built from** targets is collected.

ReportDocument
reports on
Component
built from
Component

If the *Components* **built in** the *Component* **reported on** by the *ReportDocument* are not the ones desired for a particular document, a different selection of *Components* can be identified by using the *ReportDocument* **includes** relationship. If the **includes** relationship is used, the primary path described above will not be used to collect *Components*.

ReportDocument
includes
< any *Component in the model* >

If the Autocreate *ReportBlocks* which use the **Build From List are** to operate correctly, no more than one *Component* on any branch of the component hierarchy should be linked with the **includes** relationship.

REPORTBLOCK USAGE

The following table lists the pre-defined Autocreate *ReportBlocks*. Also listed are the 490A/2167A and 498 Document Sections Templates which reference each *ReportBlock*.

TABLE B-1 Autocreate ReportBlocks Usage

ReportBlock	490A / 2167A	498
Capabilities:		
>@ - Only Allocated/Leaf Functions		
>@ - Only Allocated/Leaf Functions, Rqmts	B1/B2	
>@ - Thru Allocated/Leaf Functions	B1/B2	
>@ - Thru Leaf Functions	SRS	SRS
>@ - Thru Leaf Functions, RNets		
>@ PCs	SSS	
>@ PCs - Capabilities Only		SSS
>@ PCs - States/Modes Only		SSS/SRS
>@ PCs, Fcns		
>@ PCs, Fcns - Capabilities Only		
>@ PCs, Fcns, Rqmts		
>@ PCs, Fcns, Rqmts - Capabilities Only		
>@ PCs, Rqmts		
>@ PCs, Rqmts - Capabilities Only		
Non-functional Requirements:		
>@ Categories	SSS/B1/B2/SRS	SSS
>@ Categories, Rqmts		
>@ Categorized Rqmts		SRS
Interfaces:		
>@ External Interfaces	SSS/B1//B2/SSDD/SRS	SSS/SSDD/SRS
>@ External Interfaces, ILs	SRS	
>@ External Interfaces, ILs, Items		
>@ Internal Interfaces	B1/SSDD	
>@ Internal Interfaces, ILs		
>@ Internal Interfaces, ILs, Items		SSDD
>@ IRS Interfaces	IRS	IRS
>@ SRS Data Elements External	SRS	
>@ SRS Data Elements Internal	SRS	SRS
>@ SSDD CSCI-to-CSCI Interfaces	SSDD	

TABLE B-1 Autocreate ReportBlocks Usage (Continued)

ReportBlock	490A / 2167A	498
>@ SSDD CSCI-to-CSCI Interfaces w Items		
>@ SSDD HWCI-to-CSCI Interfaces	SSDD	
>@ SSDD HWCI-to-CSCI Interfaces w Items		
>@ SSDD HWCI-to-HWCI Interfaces	SSDD	
>@ SSDD HWCI-to-HWCI Interfaces w Items		
Allocation:		
>@ Major Component Characteristics	B1	
>@ Major Component Characteristics w Fcn I/Fs		
>@ SSDD CSCI Identification	SSDD	SSDD
>@ SSDD CSCI Identification w Items		
>@ SSDD HWCI Identification	SSDD	SSDD
>@ SSDD HWCI Identification w Items		
>@ SSDD Manual Operations Identification	SSDD	SSDD
>@ SSDD Processing Resources	SSDD	
>@ SSS Subordinate Elements	SSS	
Miscellaneous:		
>@ generic[1]		
>@ TextBlocks[2]		
>@ VRs	SSS/B1/B2/SRS	SSS/IRS?/SRS

1. Default Autocreate *ReportBlock*.
2. Available for use in any template.

REPORTBLOCK DEFINITIONS

>@ - ONLY ALLOCATED/LEAF FUNCTIONS

This *ReportBlock* creates *Sections* for allocated or leaf-level *Functions* of a function hierarchy based on a starting *Function*. A branch of the function hierarchy is traversed until either an allocated or leaf *Function* is found.

The starting *Function* is defined by one of three relationship paths:

- ❑ **used by** the current *Section*
- ❑ **performed by the** *Component* **reported by the** *ReportDocument*.
- ❑ **allocated to** the *Component* **reported by** the *ReportDocument*

An allocated *Function* is a *TimeFunction* or *DiscreteFunction* that is **allocated to** a *Component* in the **Build From List**. A leaf *Function* is either a *DiscreteFunction* or a *TimeFunction* which either has no current decomposition or it's decomposition is empty.

>@ - ONLY ALLOCATED/LEAF FUNCTIONS, RQMTS

This *ReportBlock* creates *Sections* for allocated or leaf-level *Functions* of a function hierarchy based on a starting *Function*. A branch of the function hierarchy is traversed until either an allocated or leaf *Function* is found.

Lower level *Sections* are created for *Requirements* **specified by** an allocated or leaf *Function*.

The starting *Function* is defined by one of three relationship paths:

- ❑ **used by** the current *Section*
- ❑ **performed by the** *Component* **reported by the** *ReportDocument*.
- ❑ **allocated to** the *Component* **reported by** the *ReportDocument*

An allocated *Function* is a *TimeFunction* or *DiscreteFunction* that is **allocated to** a *Component* in the **Build From List**. A leaf *Function* is either a *DiscreteFunction* or a *TimeFunction* which either has no current decomposition or it's decomposition is empty.

>@ - THRU ALLOCATED/LEAF FUNCTIONS

This *ReportBlock* creates *Sections* for all *Functions* in a function hierarchy based on a starting *Function* down to a *Function* that is **allocated to** a *Component* in the **BuildFromList** or a *Function* that is a leaf of the function hierarchy. A branch of the function hierarchy is traversed until either an allocated or leaf *Function* is found.

The starting *Function* is defined by one of three relationship paths:

- ❑ **used by** the current *Section*
- ❑ **performed by the** *Component* **reported by the** *ReportDocument*.
- ❑ **allocated to** the *Component* **reported by** the *ReportDocument*

If the **performed by** path is used, a *Section* will be not be created for the starting *Function*.

An allocated *Function* is a *TimeFunction* or *DiscreteFunction* that is **allocated to** a *Component* in the **Build From List**. A leaf *Function* is either a *DiscreteFunction* or a *TimeFunction* which either has no current decomposition or it's decomposition is empty.

>@ - THRU LEAF FUNCTIONS

This *ReportBlock* creates *Sections* for all *TimeFunctions* and *DiscreteFunctions* in a function hierarchy based on a starting *Function*.

The starting *Function* is defined by one of three relationship paths:

- ❑ **used by** the current *Section*
- ❑ **performed by the** *Component* **reported by the** *ReportDocument*.
- ❑ **allocated to** the *Component* **reported by** the *ReportDocument*

>@ - THRU LEAF FUNCTIONS, RNETS

This *ReportBlock* creates *Sections* for all *TimeFunctions, DiscreteFunctions*, and *Transforms* in a function hierarchy based on a starting *Function*.

The starting *Function* is defined by one of three relationship paths:

- ❑ **used by** the current *Section*
- ❑ **performed by the** *Component* **reported by the** *ReportDocument*.
- ❑ **allocated to** the *Component* **reported by** the *ReportDocument*

If the **performed by** path is used, a *Section* will be not be created for the starting *Function*.

>@ CATEGORIES

This *ReportBlock* creates a *Section* for each *Category* (except the top *Category*) in a *Category* hierarchy linked to the *Section* with the **uses** *Category* relationship and links the created *Sections*, with the **describes** relationship, to the *Requirements* **categorized by** the corresponding *Category*. Note that any *Requirements* **categorized by** the top *Category* are linked with the **described by** relationship to the parent *Section* (i.e., the *Section* which **uses** the top *Category)*.

- *ReportDocument*
  - **contains**
    - *Section*
      - **uses**
        - *Category*
          - **incorporates**
            - *Category* (repeated)
          - **categorizes**
            - *Requirement*

When determining the order of the *Sections* to be created, the *Categories* are sorted by Number, then by name.

The default print may be used with the created *Sections*.

>@ CATEGORIES, RQMTS

This *ReportBlock* creates a *Section* for each *Category* and *Requirement* in a Category hierarchy (except the top *Category*) linked to the *Section* with the **uses** *Category* relationship. Note that the *Sections* created for *Categories* are empty paragraphs (i.e., a *Section* has no **describes** *Category* relationship).

```
ReportDocument
    contains
        Section
            uses
                Category
                    incorporates
                        Category (repeated)
                    categorizes
                        Requirement
```

Each *Category* in the hierarchy may categorize one or more *Requirements*. If a *Category* both **categorizes** *Requirements* and **incorporates** *Categories*, the *Requirements Sections* are created after creating any subordinate *Category Sections*. When determining the order of the *Sections* to be created, *Requirements* and *Categories* are each sorted by Number, then by name.

The default print may be used with the created *Sections*.

>@ CATEGORIZED RQMTS

This *ReportBlock* creates *Sections* for required non-functional quantities, or limitations on the *Component* (i.e. non-functional requirements). The parent *Section* **uses** a *Category* that **categorizes** *Requirements*.

A *Requirement* **categorized by** a *Category* can incorporate lower-level, more detailed *Requirements*, or, through *CriticalIssues* and *Decisions*, be traced to other *Requirements*.

Each branch of the *Requirement* hierarchy is followed until the *Requirement* **specifies** a *Component* that is in the **Build From List** or the *Requirement* is a leaf in which case it has neither the **incorporates** nor **traces to** relationship.

>@ EXTERNAL INTERFACES

This *ReportBlock* creates a *Section* for each *Interface* which is either **connected thru** the *Component* **reported on** by the *ReportDocument* and/or **contains** an *ItemLink* which either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*. Any *Interface*/*ItemLink* which is linked to a *Component* with Component Type = `Environment` is ignored. When determining the order of the *Sections* to be created, the *Interfaces* are sorted by Number, then by name.

If the current *ReportDocument* has Format Standard = `498`, the following single *Section* will be created before the *Interface Sections*:

- ❑ *Section* Base.1 Interface Identification and Diagrams

where Base is the Number of the parent *Section*, usually `3.3`, which references this autocreate *ReportBlock*. The Report Block Name attribute of this created *Section* will automatically be set to `>Print External Interface Identification`.

>@ EXTERNAL INTERFACES, ILS

This *ReportBlock* creates a *Section* for each *Interface* which is either **connected thru** the *Component* **reported on** by the *ReportDocument* and/or **contains** an *ItemLink* which either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*. For each of these *Interfaces*, a lower-level *Section* is created for each *ItemLink* which **is contained by** the *Interface* and also either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*. Any *Interface*/*ItemLink* which is linked to a *Component* with Component Type = `Environment` is ignored. When determining the order of the *Sections* to be created, elements are sorted by Number, then by name.

If the current *ReportDocument* has Format Standard = `498`, the following single *Section* will be created before the *Interface Sections*:

- ❑ *Section* Base.1 Interface Identification and Diagrams

where Base is the Number of the parent *Section*, usually `3.3`, which references this autocreate *ReportBlock*. The Report Block Name attribute of this created *Section* will automatically be set to `>Print External Interface Identification`.

>@ EXTERNAL INTERFACES, ILS, ITEMS

This *ReportBlock* creates a *Section* for each *Interface* which is either **connected thru** the *Component* **reported on** by the *ReportDocument* and/or **contains** an *ItemLink* which either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*. For each of these *Interfaces*, a lower-level *Section* is created for each *ItemLink* which **is contained by** the *Interface* and also either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*. For each of these *ItemLinks*, a lower-level *Section* is created for each *TimeItem*/*DiscreteItem* **carried by** the *ItemLink*. Any *Interface*/*ItemLink* which is linked to a *Component* with Component Type = `Environment` is ignored. When determining the order of the *Sections* to be created, elements are sorted by Number, then by name.

If the current *ReportDocument* has Format Standard = `498`, the following single *Section* will be created before the *Interface Sections*:

- ❑ *Section* Base.1 Interface Identification and Diagrams

where Base is the Number of the parent *Section*, usually `3.3`, which references this autocreate *ReportBlock*. The Report Block Name attribute of this created *Section* will automatically be set to `>Print External Interface Identification`.

>@ GENERIC

This is the default Autocreate *ReportBlock*. This *ReportBlock* creates *Sections* based on a *Section* that **uses** an engineering element. This *ReportBlock* is used when the Autocreate/Output Subsection Blocks attribute of the *Section* does not specify an Autocreate *ReportBlock* name.

>@ INTERNAL INTERFACES

This *ReportBlock* creates a *Section* for each *Interface* which is **connected thru** a *Component* in the **Build From List** and **connects to** *Components* only on the **Build From List** and/or descendents of the **Build From List** elements. When determining the order of the *Sections* to be created, the *Interfaces* are sorted by Number, then by name.

If the current *ReportDocument* has Format Standard = `498`, a single *Section* will be automatically created before the *Interface Sections*:

- ❑ *Section* Base.1 Interface Identification and Diagrams

where Base is the Number of the parent *Section*, usually `3.3`, which references this autocreate *ReportBlock*. The Report Block Name attribute of this created *Section* will automatically be set to `>Print Internal Interface Identification`.

>@ INTERNAL INTERFACES, ILS

This *ReportBlock* creates a *Section* for each *Interface* which is **connected thru** a *Component* in the **Build From List** and **connects to** *Components* only on the **Build From List** and/or descendents of the **Build From List** elements. For each of these *Interfaces*, the *ReportBlock* creates *Sections* for all **contains** *ItemLinks*. When determining the order of the *Sections* to be created, elements are sorted by Number, then by name.

If the current *ReportDocument* has Format Standard = `498`, a single *Section* will be automatically created before the *Interface Sections*:

- ❑ *Section* Base.1 Interface Identification and Diagrams

where Base is the Number of the parent *Section*, usually `3.3`, which references this autocreate *ReportBlock*. The Report Block Name attribute of this created *Section* will automatically be set to `>Print Internal Interface Identification`.

>@ INTERNAL INTERFACES, ILS, ITEMS

This *ReportBlock* creates a *Section* for each *Interface* which is **connected thru** a *Component* in the **Build From List** and **connects to** *Components* only on the **Build From List** and/or descendents of the **Build From List** elements. For each of these *Interfaces*, the *ReportBlock* creates *Sections* for all **contains** *ItemLinks*. For each of these *ItemLinks*, the *ReportBlock* creates *Sections* for all **carries** *TimeItems*/*DiscreteItems*. When determining the order of the *Sections* to be created, elements are sorted by Number, then by name.

If the current *ReportDocument* has Format Standard = `498`, a single *Section* will be automatically created before the *Interface Sections*:

- ❑ *Section* Base.1 Interface Identification and Diagrams

where Base is the Number of the parent *Section*, usually `3.3`, which references this autocreate *ReportBlock*. The Report Block Name attribute of this created *Section* will automatically be set to `>Print Internal Interface Identification`.

>@ IRS INTERFACES

This *ReportBlock* creates four levels of *Sections*. A *Section* is created for each interface represented by an *ItemLink* that **enters**/**exits** a *Component* **reported by** the *ReportDocument* and does not link to a *Component* with Component Type = `Environment`. If the *ReportDocument* **includes** *Components*, the *ItemLink* must link to another element which either is **reported by** or **is included in** the *ReportDocument*. The *ItemLink Sections* are grouped below *Sections* created for each *Interface* which **contains** an included *Item-*

Link. For each *Section* created for an *ItemLink*, two lower levels of *Sections* are created for associated *Requirements*/*TimeItems*/*DiscreteItems*.

- *Section* Base.1 section for interface diagram
- *Section* Base.x individual *Sections* for each *Interface*
- *Section* Base.x.y individual *Sections* for each *ItemLink*
- *Section* Base.x.y.1 groups interface requirements defined by *ItemLink* **specified by** *Requirement*.
- *Section* Base.x.y.1.z individual *Sections* for each *Requirement*
- *Section* Base.x.y.2 groups data requirements defined by *ItemLink* **carries** *TimeItem/DiscreteItem*.
- *Section* Base.x.y.2.z individual *Sections* for each *TimeItem/DiscreteItem.*

where Base is the Number, usually 3, of the parent *Section* referencing the autocreate *ReportBlock*. If the current *ReportDocument* has the attribute Format Standard = `498` a final *Section* with a number Base.x will be created for requirements precedence and criticality information.

When determining the order of the *Sections* to be created, *Interfaces*, *ItemLinks* and *Requirements* are sorted by Number, then by name and *TimeItems*/*DiscreteItems* are sorted by name.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.1 `>Print IRS Interface Diagrams`
- Base.x `>Print Interface, Rqmts, ItemLink Table`
- Base.x.y.2 `>Print IRS ItemLink Message Table`
- Base.x.y.2.z `>Print Item Definition, Rqmts.`

If a final precedence and criticality *Section* is created, the Report Block Name will be unassigned.

>@ MAJOR COMPONENT CHARACTERISTICS

This *ReportBlock* creates a *Section* for each *Component* in the **BuildFromList**. Each *Section* has two lower level *Sections* created automatically:

- *Section* Base.x.1 Performance Characteristics
- *Section* Base.x.2 Physical Characteristics

where Base is the Number, usually `3.7`, of the parent *Section* referencing the autocreate *ReportBlock*. Each autocreated *Section* **describes** the *Component*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.x.1 `>Print Major Component Performance Characteristics`
- Base.x.2 `>Print Major Component Physical Characteristics`

>@ MAJOR COMPONENT CHARACTERISTICS W FCN IFS

This *ReportBlock* creates a *Section* for each *Component* in the **BuildFromList**. Each *Section* has three lower level *Sections* created automatically:

- *Section* Base.x.1 Performance Characteristics
- *Section* Base.x.2 Physical Characteristics
- *Section* Base.x.3 Component Interfaces

where Base is the Number, usually `3.7`, of the parent *Section* referencing the autocreate *ReportBlock*. Each autocreated *Section* **describes** the *Component*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.x.1 `>Print Major Component Performance Characteristics`
- Base.x.2 `>Print Major Component Physical Characteristics`
- Base.x.3 `>Print Major Component Functional Interfaces`

>@ PCS

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*.

The Output Sections *ReportBlock* `>Print PCs` or `>Print PCs, Fcns` must be used in order to print the autocreated *Sections*.

>@ PCS - CAPABILITIES ONLY

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* with Type = `Capability` in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*.

The Output Sections *ReportBlock* `>Print PCs` or `>Print PCs, Fcns` must be used in order to print the autocreated *Sections*.

>@ PCS - STATES/MODES ONLY

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* with Type = `State` or `Mode` in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*.

The Output Sections *ReportBlock* `>Print PCs` or `>Print PCs, Fcns` must be used in order to print the autocreated *Sections*.

>@ PCS, FCNS

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*.

Lower level *Sections* are created for each *Function* that motivates a leaf-level *PerformanceCharacteristic*. The Report Block Name attribute of the *Sections* autocreated for *Functions* is automatically set to `>Print Capability Function, Rqmts`.

The Output Sections *ReportBlock* `>Print PCs` must be used in order to print the *Requirements* linked directly to the *PerformanceCharacteristics* for which this Autocreate Block creates *Sections*.

>@ PCS, FCNS - CAPABILITIES ONLY

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* with Type = `Capability` in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*.

Lower level *Sections* are created for each *Function* that motivates a leaf-level *PerformanceCharacteristic*. The Report Block Name attribute of the *Sections* autocreated for *Functions* is automatically set to `>Print Capability Function, Rqmts`.

The Output Sections *ReportBlock* `>Print PCs` must be used in order to print the *Requirements* linked directly to the *PerformanceCharacteristics* for which this Autocreate Block creates *Sections*.

>@ PCS, FCNS, RQMTS

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*

- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic.*

Lower level *Sections* are created for each *Function* that **motivates** a leaf-level *PerformanceCharacteristic*. Below the *Function* level, a *Section* is created for each *Requirement* that **specifies** the *Function* unless that *Requirement* **specifies** another *PerformanceCharacteristic* in the characteristics hierarchy and does not also specify the current *PerformanceCharacteristic*. The Report Block Name attribute of the *Sections* autocreated for *Functions* and *Requirements* is unassigned.

If a leaf-level *PerformanceCharacteristic* is not **motivated by** a *Function*, *Sections* are created for each *Requirement* which **specifies** the *PerformanceCharacteristic.*

The default print can be used with this Autocreate Block if only the leaf *PerformanceCharacteristics* are **specified by** *Requirements*; otherwise, the Output Sections *ReportBlock* `>Print PCs` must be used.

>@ PCS, FCNS, RQMTS - CAPABILITIES ONLY

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* with Type = `Capability` in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic.*

Lower level *Sections* are created for each *Function* that **motivates** a leaf-level *PerformanceCharacteristic*. Below the *Function* level, a *Section* is created for each *Requirement* that **specifies** the *Function* unless that *Requirement* **specifies** another *PerformanceCharacteristic* in the characteristics hierarchy and does not also specify the current *PerformanceCharacteristic*. The Report Block Name attribute of the *Sections* autocreated for *Functions* and *Requirements* is unassigned.

If a leaf-level *PerformanceCharacteristic* is not **motivated by** a *Function*, *Sections* are created for each *Requirement* which **specifies** the *PerformanceCharacteristic.*

The default print can be used with this Autocreate Block if only the leaf *PerformanceCharacteristics* are **specified by** *Requirements*; otherwise, the Output Sections *ReportBlock* `>Print PCs` must be used.

>@ PCS, RQMTS

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic.* The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*

Lower level *Sections* are created for *Requirements* linked directly or indirectly to the leaf-level *PerformanceCharacteristics.*

PerformanceCharacteristic
  specified by

Requirement
motivated by
TimeFunction, *DiscreteFunction*
specified by
Requirement

Sections are created for each *Requirement* which **specifies** a leaf *PerformanceCharacteristic*. If a leaf-level *PerformanceCharacteristic* is **motivated by** one or more *TimeFunctions* or *DiscreteFunctions*, a *Section* is created for each *Requirement* which **specifies** the *Function* unless that *Requirement* **specifies** another *PerformanceCharacteristic* in the characteristics hierarchy and does not also specify the current *PerformanceCharacteristic*.

The default print can be used with this Autocreate Block if only the leaf *PerformanceCharacteristics* are **specified by** *Requirements*; otherwise, the Output Sections *ReportBlock* `>Print PCs` must be used.

>@ PCs, Rqmts - Capabilities Only

This *ReportBlock* creates a *Section* for each *PerformanceCharacteristic* with Type = `Capability` in a performance characteristics hierarchy based on a starting *PerformanceCharacteristic*. The starting *PerformanceCharacteristic(s)* is identified by either of two relationship paths:

- ❑ *ReportDocument* **contains** *Section* **uses** *PerformanceCharacteristic*
- ❑ *ReportDocument* **reports on** *Component* **exhibits** *PerformanceCharacteristic*

Lower level *Sections* are created for *Requirements* linked directly or indirectly to the leaf-level *PerformanceCharacteristics*.

PerformanceCharacteristic
specified by
Requirement
motivated by
TimeFunction, *DiscreteFunction*
specified by
Requirement

Sections are created for each *Requirement* which **specifies** a leaf *PerformanceCharacteristic*. If a leaf-level *PerformanceCharacteristic* is **motivated by** one or more *TimeFunctions* or *DiscreteFunctions*, a *Section* is created for each *Requirement* which **specifies** the *Function* unless that *Requirement* **specifies** another *PerformanceCharacteristic* in the characteristics hierarchy and does not also specify the current *PerformanceCharacteristic*.

The default print can be used with this Autocreate Block if only the leaf *PerformanceCharacteristics* are **specified by** *Requirements*; otherwise, the Output Sections *ReportBlock* `>Print PCs` must be used.

>@ SRS Data Elements External

This *ReportBlock* creates a *Section* for each *TimeItem*/*DiscreteItem* that is **carried by** an *ItemLink* that either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*. An *ItemLink* which is linked to a *Component* with Component Type = `Environment` is ignored. When determining the order of the *Sections* to be created, elements are sorted by Number, then by name.

>@ SRS DATA ELEMENTS INTERNAL

This *ReportBlock* creates a *Section* for each *TimeItem/DiscreteItem* which is both **input to** and **output from** a **Leaf Capability Function**. A **Leaf Capability Function** is either a *TimeFunction* **described by** a leaf-level *Section* with Number =`3.2.*` that **is contained by** the *ReportDocument* or a *DiscreteFunction* **described by** a *Section* with Number =`3.2.*` that **is contained by** the *ReportDocument*.

>@ SSDD CSCI IDENTIFICATION

This *ReportBlock* creates a *Section* for each of the *Components* of Component Type = `CSCI` or `FWCI` in the **Build From List**. Each of these *Sections* **describes** the corresponding *Component.* This *ReportBlock* autocreates different lower level *Sections* depending on the *ReportDocument* Format Standard attribute.

490A/2167A

For a 490A/2167A template (i.e., Format Standard = `490A/2167A` or `nil`), each *Component Section* has three lower level *Sections* created automatically:

- *Section* Base.x.1 Allocated Requirements
- *Section* Base.x.2 Design Constraints
- *Section* Base.x.3 System External Interfaces

where Base is the Number, usually 4.2, of the parent *Section* referencing the Autocreate *ReportBlock*. Each autocreated Allocated Requirements and Design Constraints *Section* **describes** the *Component.*

Sections are created below a System External Interfaces *Section* for each external *Interface* and **contains** *ItemLink* addressed by the *Component.* An *Interface* is addressed by a *Component* if one or more of the following conditions is true:

- the *Interface* **connects thru** the *Component* and **connects to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants
- the *Interface* **contains** an *ItemLink* that **enters** the *Component* and is **output from** at least one *Component* that is not a *Component* on the **BuildFromList** or one of its descendants
- the *Interface* **contains** an *ItemLink* that **exits** the *Component* and is **input to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants.

Each of the autocreated *Sections* **describes** the *Interface* or *ItemLink*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.x.1 `>Print SSDD CI Allocated Requirements`
- Base.x.2 `>Print SSDD CI Design Constraints`

- ❑ Base.x.3.y `>Print SSDD CI System External Interfaces`
- ❑ Base.x.3.y.z `>Print SSDD CI System External Interfaces.`

498

If the Format Standard = `498`, each *Component Section* has two lower level *Sections* created automatically:

- ❑ *Section* Base.x.1 Allocated Requirements
- ❑ *Section* Base.x.2 Design Constraints

where Base is the Number, usually 4.1.2, of the parent *Section* referencing the Autocreate *ReportBlock*. Each autocreated Allocated Requirements and Design Constraints *Section* **describes** the *Component*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- ❑ Base.x.1 `>Print SSDD CI Allocated Requirements`
- ❑ Base.x.2 `>Print SSDD CI Design Constraints`

>@ SSDD CSCI IDENTIFICATION W ITEMS

This *ReportBlock* creates a *Section* for each of the *Components* of Component Type = `CSCI` or `FWCI` in the **Build From List**. Each of these *Sections* **describes** the corresponding *Component*.

This *ReportBlock* is intended for use in 490A/2167A templates. If used in a 498 template, it will autocreate according to the 498 description of `>@ CSCI Subordinate Elements for SSDD`.

For a 490A/2167A template, each *Component Section* has three lower level *Sections* created automatically:

- ❑ *Section* Base.x.1 Allocated Requirements
- ❑ *Section* Base.x.2 Design Constraints
- ❑ *Section* Base.x.3 System External Interfaces

where Base is the number, usually 4.2, of the parent *Section* referencing the Autocreate *ReportBlock*.

Each autocreated Allocated Requirements and Design Constraints *Section* **describes** the *Component*.

Sections are created below a System External Interfaces *Section* for each external *Interface* and **contains** *ItemLink* addressed by the *Component*. An *Interface* is addressed by a *Component* if one or more of the following conditions is true:

- ❑ the *Interface* **connects thru** the *Component* and **connects to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants
- ❑ the *Interface* **contains** an *ItemLink* that **enters** the *Component* and is **output from** at least one *Component* that is not a *Component* on the **Build From List** or one of its

descendants

- the *Interface* **contains** an *ItemLink* that **exits** the *Component* and is **input to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants

Sections are created below an *ItemLink Section* for each *TimeItem*/*DiscreteItem* **carried by** the *ItemLink*. Each of the autocreated *Sections* **describes** the *Interface*, *ItemLink* or *TimeItem*/*DiscreteItem*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.x.1 `>Print SSDD CI Allocated Requirements`
- Base.x.2 `>Print SSDD CI Design Constraints`.

>@ SSDD CSCI-TO-CSCI INTERFACES

This *ReportBlock* creates a *Section* for each *Interface* **connected thru** two or more *Components* in the **BuildFromList** with ComponentType = `CSCI` or `FWCI`. If the *Interface* **contains** *ItemLinks*, a *Section* is created for each *ItemLink.*

>@ SSDD CSCI-TO-CSCI INTERFACES W ITEMS

This *ReportBlock* creates a *Section* for each *Interface* **connected thru** two or more *Components* in the **BuildFromList** with ComponentType = `CSCI` or `FWCI`. If the *Interface* **contains** *ItemLinks*, a *Section* is created for each *ItemLink.* If an *ItemLink* **carries** *TimeItems*/*DiscreteItems*, a *Section* is created for each *Item*.

>@ SSDD HWCI IDENTIFICATION

This *ReportBlock* creates a *Section* for each of the *Components* of Component Type = `HWCI` in the **Build From List**. Each of these *Sections* **describes** the corresponding *Component*. This *ReportBlock* autocreates different lower level *Sections* depending on the *ReportDocument* Format Standard attribute.

490A/2167A

For a 490A/2167A template (i.e., Format Standard = `490A/2167A` or `nil`), each *Component Section* has three lower level *Sections* created automatically:

- *Section* Base.x.1 Allocated Requirements
- *Section* Base.x.2 Design Constraints
- *Section* Base.x.3 System External Interfaces

where Base is the number, usually 4.1, of the parent *Section* referencing the Autocreate *ReportBlock*. Each autocreated Allocated Requirements and Design Constraints *Section* **describes** the *Component*.

Sections are created below a System External Interfaces *Section* for each external *Interface* and **contains** *ItemLink* addressed by the *Component*. An *Interface* is addressed by a *Component* if one or more of the following conditions is true:

- the *Interface* **connects thru** the *Component* and **connects to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants
- the *Interface* **contains** an *ItemLink* that **enters** the *Component* and is **output from** at least one *Component* that is not a *Component* on the **BuildFromList** or one of its descendants
- the *Interface* **contains** an *ItemLink* that **exits** the *Component* and is **input to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants

Each of the autocreated *Sections* **describes** the *Interface* or *ItemLink*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed where Base is the Number of the parent *Section* referencing the autocreate *ReportBlock*:

- Base.x.1 `>Print SSDD CI Allocated Requirements`
- Base.x.2 `>Print SSDD CI Design Constraints`
- Base.x.3.y `>Print SSDD CI System External Interfaces`
- Base.x.3.y.z `>Print SSDD CI System External Interfaces.`

498

For a 498 template (i.e., Format Standard = `498)`, each *Component Section* has two lower level *Sections* created automatically:

- *Section* Base.x.1 Allocated Requirements
- *Section* Base.x.2 Design Constraints

where Base is the number, usually 4.1.1, of the parent *Section* referencing the Autocreate *ReportBlock*.

If the `HWCI` *Component* **executes** a *Component* that is both on the **Build From List** and has Component Type = `CSCI` or `FWCI`, then a third lower level Section is created automatically:

- *Section* Base.x.3 Resource Utilization and Characteristics.

Each autocreated Allocated Requirements, Design Constraints and Resource *Section* **describes** the *Component.*

Sections are created below a Resource Utilization and Characteristics *Section* for each *PerformanceIndex* **exhibited by** the *Component.*

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed where Base is the Number of the parent *Section* referencing the autocreate *ReportBlock*:

- Base.x.1 `>Print SSDD CI Allocated Requirements`
- Base.x.2 `>Print SSDD CI Design Constraints`

- Base.x.3.y `>Print SSDD CI Utilization Table.`

>@ SSDD HWCI IDENTIFICATION W ITEMS

This *ReportBlock* creates a *Section* for each of the *Components* of Component Type = `HWCI` in the **Build From List**. Each of these *Sections* **describes** the corresponding *Component*.

This *ReportBlock* is intended for use in 490A/2167A templates. If used in a 498 template, it will autocreate lower level Sections as documented under the 498 description of `>@ HWCI Subordinate Elements for SSDD`.

For a 490A/2167A template, each *Section* has three lower level *Sections* created automatically:

- *Section* Base.x.1 Allocated Requirements
- *Section* Base.x.2 Design Constraints
- *Section* Base.x.3 System External Interfaces

where Base is the number, usually 4.1, of the parent *Section* referencing the Autocreate *ReportBlock*.

Each autocreated Allocated Requirements and Design Constraints *Section* **describes** the *Component*.

Sections are created below a System External Interfaces *Section* for each external *Interface* and **contains** *ItemLink* addressed by the *Component*. An *Interface* is addressed by a *Component* if one or more of the following conditions is true:

- the *Interface* **connects thru** the *Component* and **connects to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants
- the *Interface* **contains** an *ItemLink* that **enters** the *Component* and is **output from** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants
- the *Interface* **contains** an *ItemLink* that **exits** the *Component* and is **input to** at least one *Component* that is not a *Component* on the **Build From List** or one of its descendants

Sections are created below an *ItemLink Section* for each *TimeItem*/*DiscreteItem* **carried by** the *ItemLink*. Each of the autocreated *Sections* **describes** the *Interface*, *ItemLink* or *TimeItem*/*DiscreteItem*.

This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.x.1 `>Print SSDD CI Allocated Requirements`
- Base.x.2 `>Print SSDD CI Design Constraints`.

>@ SSDD HWCI TO-CSCI INTERFACES

This *ReportBlock* creates a *Section* for each *Interface* **connected thru** two or more *Components* in the **Build From List** with Component Type = `HWCI` or `CSCI` or `FWCI`. The *Interface* must be **connected thru** at least one *Component* with ComponentType = `HWCI` and one *Component* with ComponentType = `CSCI` or `FWCI`. If the *Interface* **contains** *ItemLinks*, a *Section* is created for each *ItemLink.*

>@ SSDD HWCI-TO-CSCI INTERFACES W ITEMS

This *ReportBlock* creates a *Section* for each *Interface* **connected thru** two or more *Components* in the **Build From List** with Component Type = `HWCI` or `CSCI` or `FWCI`. The *Interface* must be **connected thru** at least one *Component* with ComponentType = `HWCI` and one *Component* with ComponentType = `CSCI` or `FWCI`. If the *Interface* **contains** *ItemLinks*, a *Section* is created for each *ItemLink.* If an *ItemLink* **carries** *TimeItems*/*DiscreteItems*, a *Section* is created for each *Item*.

>@ SSDD HWCI-TO-HWCI INTERFACES

This *ReportBlock* creates a *Section* for each *Interface* **connected thru** two or more *Components* in the **Build From List** with Component Type = `HWCI`. If the *Interface* **contains** *ItemLinks*, a *Section* is created for each *ItemLink.*

>@ SSDD HWCI-TO-HWCI INTERFACES W ITEMS

This *ReportBlock* creates a *Section* for each *Interface* **connected thru** two or more *Components* in the **Build From List** with Component Type = `HWCI`. If the *Interface* **contains** *ItemLinks*, a *Section* is created for each *ItemLink.* If an *ItemLink* **carries** *TimeItems*/ *DiscreteItems*, a *Section* is created for each *Item*.

>@ SSDD MANUAL OPERATIONS IDENTIFICATION

This *ReportBlock* creates a *Section* for each of the *Components* of Component Type = `Human` in the **Build From List**. Each of these *Sections* **describes** the corresponding *Component* and has three lower level *Sections* created automatically:

- *Section* Base.x.1 Allocated Requirements
- *Section* Base.x.2 Design Constraints

where Base is the Number of the parent *Section* referencing the Autocreate *ReportBlock*.

Each autocreated *Section* **describes** the *Component*. This Autocreate *ReportBlock* automatically assigns the following Output *ReportBlock* names to the Report Block Name attribute of each *Section* that it creates with a number of the type listed:

- Base.x.1 `>Print SSDD CI Allocated Requirements`
- Base.x.2 `>Print SSDD CI Design Constraints`.

>@ SSDD PROCESSING RESOURCES

This *ReportBlock* creates a *Section* for each *Component* that **executes** a *Component* in the **Build From List** with ComponentType = `CSCI` or `FWCI`. A lower level *Section* is created for each *PerformanceIndex* **exhibited by** the *Component*.

>@ SSS SUBORDINATE ELEMENTS

This *ReportBlock* creates a *Section* for each *Component* in the **Build From List**.

>@ TEXTBLOCKS

This *ReportBlock* creates a *Section* for each *TextBlock* that is **used by** the current *Section*. The *TextBlocks* are listed ordered by Number and then by name.

>@ VRS

This *ReportBlock* creates *Sections* for *VerificationRequirements.* A *Section* is created if the *VerificationRequirement* **verifies** a *Requirement* that either is **described by** a *Section;* or **specifies** a *TimeFunction*/*DiscreteFunction* which either is **described by** a *Section* or **motivates** a *PerformanceCharacteristic* **described by** a *Section*; or **specifies** a *PerformanceCharacteristic* **described by** a *Section*. In addition, the *Section* must have the **is contained by** relationship to the current *ReportDocument* and have a Number that adheres to the **Traceability List** rules.

Appendix

C

Output Sections Report Blocks

This appendix describes the predefined Output *ReportBlocks* provided with the Design Guide Type R reports. These *ReportBlocks* are used with a *ReportDocument*, a set of manually and automatically defined *Sections* and the Output Sections Report to produce a document based on the engineering elements in the SDD.

Common attribute, relationship and target element type requirements are defined elsewhere in this document. The individual Output *ReportBlock* definitions specify only attributes, relationships and target element types unique to that *ReportBlock*. The definitions are arranged in alphabetical order.

STANDARD ELEMENT PRINT INFORMATION

STANDARD ELEMENT TITLES

When an element Title is printed, it is followed by it's Project Unique ID. If the element Title is empty, the element name is printed. This printing will be referred to as the **Standard Title** in the *ReportBlock* definitions in this appendix

STANDARD ELEMENT DESCRIPTIONS

When an element Description is printed within a section, it's Project Unique ID, *Classification* and *NonRDDIllustration* information are also printed when available. This printing will be referred to as the **Standard Description**in the *ReportBlock* definitions in this appendix. The **Standard Description** will be printed as a separate paragraph within a section for all engineering elements except *Requirements*.

The printing of a *Requirement* **Standard Description** follows the **Requirement Formatting Rules** The **Requirement Formatting Rules**are governed by the *Requirements* Report Option attribute and are as follows:

- ❑ If Report Option = `begin Section` or `begin paragraph`, the **Standard Description** will be printed as a new paragraph.
- ❑ If Report Option = `begin lettered item`, the **Standard Description** will be printed as a new lettered item within the section.
- ❑ If Report Option = `follow previous text`, the *Requirement* Description followed by its Project Unique ID will be appended to the current text (i.e., either a paragraph or lettered item). The *Classification* of the *Requirement* will be compared to the current highest classification for any prior text, and the resultant highest classification will take precedence for the combined text. Any *NonRDDIllustrations* associated with the *Requirement* will be added to a list for the combined text and processed before the start of the next paragraph or lettered item. If no prior text exists, the **StandardDescription** of the *Requirement* will start a new paragraph.
- ❑ A *Requirement* **Standard Description** will not be appended to the **Standard Section Titling**.

STANDARD SECTION TITLING

When a *Section* is processed, the *Section* Number, **Standard Title** and **Standard Description** of the current *Section* and the **Standard Description** of all *TextBlocks* **described by** the *Section* are printed. This *Section* and *TextBlock* related printing will be referred to as the **Standard Section Titling** in the *ReportBlock* definitions in this appendix.

SPECIAL LISTS DEFINITIONS

BUILD FROM LIST

Each *ReportDocument* **reports on** a primary *Component*. A document may require an additional selection of *Components*, subordinate elements of the primary, to be identified. These are the *Components* to which requirements are allocated in the document.

The collecting of these allocation *Components* is done only once in the Output Sections Report, prior to the processing of all individual *Sections*. In this appendix, this collection of *Components* will be referred to as the **Build From List**.

Two different relationship paths can be used to define the **Build From List**. The primary path is *ReportDocument* **reports on** *Component* **built from** the next level of *Component* elements. The **built from** relationship is NOT followed recursively. Only the first level of **built from** targets is collected.

```
ReportDocument
    reports on
        Component
            built from
                Component
```

If the *Components* **built in** the *Component* **reported on** by the *ReportDocument* are not the ones desired for a particular document, a different selection of *Components* can be identified by using the *ReportDocument* **includes** relationship. If the **includes** relationship is used, the primary path described above will not be used to collect *Components*.

```
ReportDocument
    includes
        < any Component in the model >
```

In order for the Output *ReportBlocks* which use the **Build From List** to operate correctly, no more than one *Component* on any branch of the component hierarchy should be linked with the **includes** relationship.

TRACEABILITY LIST

A document may require a group of *Sections* to be identified in order to produce a traceability matrix. These *Sections* are identified by their Number. To be included in the **Traceability List**, a *Section* must have a Number within a range dependent on the current document Format Standard and Document Type:

- ❑ For Format Standard `490A/2167A` and
  - Document Type SRS, Number must begin with `3.*` (except `3.12` and `3.12.*`) or `5.*`
  - Document Type SSDD, Number must begin with either `3.*`, `5.*`, or `6.*`
  - All other Document Types, Number must begin with `3.*` (except `3.7` or `3.7.*`) or `5.*`
- ❑ For Format Standard `498` and all Document Types, Number must begin with `3.*`.

REPORTBLOCK USAGE

The following table lists the pre-defined Output *ReportBlocks*. Also listed are the 490A/2167A and 498 Document Sections Templates which reference each *ReportBlock*.

TABLE C-1 Output ReportBlocks Usage

ReportBlock	490A / 2167A	498
Capabilities:		
>Print Description & Behavior Diagram		
>Print Element and Rqmts Description		
>Print Function, I/O, graph and graph references	B1/B2	
>Print Function, Rqmts		
>Print Function, Rqmts, and graph		
>Print Function, Rqmts, and I/O		
>Print Function, Rqmts, I/O, and graph	SRS	SRS
>Print Function, Rqmts, I/O, graph and graph references		
>Print PCs	SSS	SSS
>Print PCs, Fcns		
>Print State, Mode, Capability/Fcn Table	SSS/SRS	SSS/SRS
>Print State, Mode, CI Table	SSDD	
Interfaces:		
>Print CSCI Internal Interfaces	SRS	SRS
>Print External Interface Identification		SSS/SSDD/SRS
>Print External Interfaces		
>Print External ItemLink, Message Table		
>Print Interfaces w Reference	SSS/B1/B2/SRS	SSS/SSDD/SRS
>Print Internal Item Usage Table		
>Print Internal Interface Identification		SSDD
>Print IRS Interface Diagrams	IRS	IRS
>Print IRS Interface Identification	IRS	IRS
>Print IRS Interface, Rqmts, ItemLink Table	IRS	IRS
>Print IRS ItemLink Message Table	IRS	IRS
>Print Item Definition, Rqmts	IRS	IRS
>Print Major Component Functional Interfaces		
>Print SSDD Internal Interfaces	SSDD	
>Print SSDD CI System External Interfaces	SSDD	
>Print SSDD System External Interfaces	SSDD	
>Print SSDD System Internal Interfaces	SSDD	

TABLE C-1 Output ReportBlocks Usage (Continued)

ReportBlock	490A / 2167A	498
Allocation:		
>Print Major Component List	B1/SSDD	SSDD
>Print Major Component Performance Characteristics	B1	
>Print Major Component Physical Characteristics	B1	
>Print Reported On Element's Component Hierarchy		
>Print Reported On Element's Component View	SSDD	
>Print SSDD CI Allocated Requirements	SSDD	SSDD
>Print SSDD CI Design Constraints	SSDD	SSDD
>Print SSDD CI Utilization Table	SSDD	SSDD
>Print Subordinate Elements Rqmts Table	SSS	
Miscellaneous:		
>Print (2.1) Government Documents List	SSS/B1/B2/SSDD/IRS/SRS	SSS/SSDD/IRS/SRS
>Print (2.2) Non-Government Documents List	SSS/B1/B2/SSDD/IRS/SRS	SSS/SSDD/IRS/SRS
>Print (Do Not Print)		
>Print Description[1]		
>Print Item Definition	SRS	SSDD/SRS
>Print Previously Referenced Paragraph		
>Print Reported On Element's Context Diagram	SSDD	
>Print Reported On Element's Description	SSDD/SRS	SSS/SSDD
>Print Reported On Element's Description & Context Diagram	SSS/B1/B2	
>Print Reported On Element's Top-Level Behavior	B1/B2	
>Print Requirements Subparagraphs Only		
>Print Requirements Traceability Matrix	SSS/B1/B2/SRS	SSS/SRS
>Print Requirements Verification Matrix	SSS/B1/B2/SRS	SSS/SRS
>Print SRS External Data Table	SRS	
>Print SRS Internal Data Table	SRS	SRS
>Print SSDD Requirements Traceability Matrix	SSDD	SSDD
>Print VRs	SSS/B1/B2/SRS	SSS/SRS

[1] Default Output *ReportBlock*.

REPORTBLOCK DEFINITIONS

>PRINT (2.1) GOVERNMENT DOCUMENTS LIST

This *ReportBlock* prints the **Standard Section Titling** and lower level sections associated with *ApplicableDocuments* that are linked to the *ReportDocument* and have Government set to `Federal`, `Military`, `Government or Other Government Agency`. The *ApplicableDocuments* are sorted by Document Number and name. They are grouped in the subsections according to Publications Type and Government values.

TABLE C-2 Grouping of Government Documents by Subsection and Publication Type

SUBSECTION	PUBLICATION TYPE ATTRIBUTE
SPECIFICATIONS	SPECIFICATIONS
STANDARDS	STANDARDS
DRAWINGS	DRAWINGS
OTHER PUBLICATIONS	Manual Regulation Handbook Bulletin Other

>PRINT (2.2) NON-GOVERNMENT DOCUMENTS LIST

This *ReportBlock* prints the **Standard Section Titling** and lower level sections associated with *ApplicableDocuments* that are linked to the *ReportDocument* and the Government attribute value is `Non-Government` or is undefined. The *ApplicableDocuments* are sorted by Document Number and name. They are grouped in the subsections according to Publications Type and Government values.

TABLE C-3 Grouping of Non-Government Documents by Subsection and Publication Type

SUBSECTION	PUBLICATION TYPE ATTRIBUTE
SPECIFICATIONS	SPECIFICATIONS
STANDARDS	STANDARDS
DRAWINGS	DRAWINGS
OTHER PUBLICATIONS	Manual Regulation Handbook Bulletin Other

>PRINT (DO NOT PRINT)

This *ReportBlock* suppresses all printing for the *Section* including the **Standard Section Titling.**

>PRINT CSCI INTERNAL INTERFACES

This *ReportBlock* prints the **Standard Section Titling** and two three column table listing CSCI internal data elements and their capability sources and destinations. Internal data elements are *TimeItems*/*DiscreteItems* which are both **input to** and **output from** leaf-level capabilities of the CSCI (i.e, leaf-level *TimeFunctions* and/or *DiscreteFunctions* **described by** a *Section* with a Number = `3.2.*` that **is contained by** the current *Report-Document)*.

The first of the two tables produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of a leaf-level capability (i.e., *TimeFunction*/*DiscreteFunction*) of the CSCI.
- Column Two: The **Standard Title** of each *TimeItem*/*DiscreteItem* **input to** the *Function* listed in Column One that is also **output from** a leaf-level capability of the CSCI or that has no source.
- Column Three: The **Standard Title** of each CSCI capability which is a source for the *Item* listed in Column Two. If there is no source capability for the *Item*, this column is blank.

The second of the two tables produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of a leaf-level capability (i.e., *TimeFunction*/*DiscreteFunction*) of the CSCI.
- Column Two: The **Standard Title** of each *TimeItem*/*DiscreteItem* **output from** the *Function* listed in Column One that is also **input to** a leaf-level capability of the CSCI or that has no destination.
- Column Three: The **Standard Title** of each CSCI capability which is a destination for the *Item* listed in Column Two. If there is no destination capability for the *Item*, this column is blank.

>PRINT DESCRIPTION

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of each engineering element **described by** the *Section*. The **Requirement Formatting Rules** are applied when printing the **Standard Description** of *Requirements* elements.

>PRINT DESCRIPTION & BEHAVIOR DIAGRAM

This *ReportBlock* prints the **Standard Section Titling**, the **Standard** Description of each element **described by** the *Section* and, if that element is a *Function*, the function’s Behavior Diagram based on the *FNet* (if it is not empty) that is the **current decomposition of** the *Function*.

>PRINT ELEMENT AND RQMTS DESCRIPTION

This *ReportBlock* prints the **Standard Section Titling,** the **Standard Description** of each element **described by** the *Section* and the **Standard Description** of every *Requirement* that **specifies** that element using the **Requirement Formatting Rules**

>PRINT EXTERNAL INTERFACE IDENTIFICATION

This *ReportBlock* prints the **Standard Section Titling** and a table listing the external *Interfaces* for the *Component* **reported on** by the current document.

The table produced by this *ReportBlock* contains the following:

- ❑ Column One: The **Standard Title** of each *Interface* that **connects thru** the **reported on** *Component*.
- ❑ Column Two: The **Standard Title** of each *Component* that **connects to** the *Interface* in Column One and that is not in the component hierarchy that begins with the **reported on** *Component.*

>PRINT EXTERNAL INTERFACES

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section* and, if that element is an *Interface*, a two column table of *ItemLinks*, *Items* and *Components* showing the interfaces between the *Component* **reported by** the *ReportDocument* and all external *Components*. The table contains an entry for each *ItemLink* which **is contained by** the *Interface* and also either **exits** or **enters** the *Component* **reported on** by the *ReportDocument*.

The table produced by this *ReportBlock* contains:

- ❑ Column One: The **Standard Title** of the *ItemLink* and the **Standard Title** of any *Items* **carried by** the *ItemLink*.
- ❑ Column Two: The **Standard Title** of the highest level external *Component* in the component hierarchy which is an ancestor of the *ItemLink* **output from**/**input to** *Component* and also is **exited by**/**entered by** the *ItemLink*.

>PRINT EXTERNAL ITEMLINK, MESSAGE TABLE

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of any elements **described by** the *Section* and, if the element is an *ItemLink*, a table listing the *TimeItems*/*DiscreteItems* **carried by** the ItemLink.

The table produced by this *ReportBlock* contains:

- ❑ Column One: The **Standard Title** of the *DiscreteItem*/*TimeItem*.
- ❑ Column Two: The Description of the *DiscreteItem*/*TimeItem* listed in Column One.

A one sentence lead-in is automatically printed before the table. This sentence lists the source and destination *Components* for the *ItemLink*. The external source/destination is the highest level external *Component* in the component hierarchy which is an ancestor of the *ItemLink* **output from**/**input to** *Component* and also is **exited by**/**entered by** the *ItemLink*.

Note: renamed from `>Print ItemLink I/F elements`.

>PRINT FUNCTION, I/O, GRAPH AND GRAPH REFERENCES

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section.* If that element is a *Function,* an Input/Output Table identifying all *TimeItems*/*DiscreteItems* **input to/output from** the *Function* is also printed. The table consists of the following:

- Column One: The **Standard Title** of an *Item* that is **input to**/**output from** the *Function*.
- Column Two: The Description of the *Item* listed in Column One.

This *ReportBlock* may also print a graph based on the *FNet*, *RNet*, or *TNet* (if it is not empty) that is the **current decomposition of** the *Function*. The graphs available are a Behavior Diagram, or FFBD, or IDEF0, or an FFBD and IDEF0. The default is to not print any graph. If a graph is to be printed, the type of graph must be selected from the report's run-time options. Note: IDEF0 and DFD are not valid graph choices for *DiscreteFunctions*. Only a Behavior Diagram can be used for *Transforms*.

If the *Function* is a *DiscreteFunction/Transform* **decomposed by** an *RNet/TNet* with a non-null decomposition and the report's run-time options to print graphs was selected, a table clarifying the graph nodes is printed before the graph. The table consists of the following:

- Column One: The element type and **Standard Title** of an element **referred by** the *RNet*/*TNet* if that element has a non-empty Description.
- Column Two: The Description of the element identified in Column One. If the element is a *Condition*, one of the following is printed if it is available:
    - The *Condition* Expression if it is not empty.
    - The name of the *ProbabilityDistribution* that **supplies value to** the *Condition*.
    - The *Condition* Exit Probability if it is not empty.

>PRINT FUNCTION, RQMTS

This *ReportBlock* prints the **Standard Section Titling,** the **Standard Description** of each element **described by** the *Section*. If that element is a *Function* and the previous *Section* has no **describes** relationship to a *PerformanceCharacteristic,* the **Standard Description** of every *Requirement* that **specifies** that *Function* is printed using the **Requirement Formatting Rules** If the previous section describes a *PerformanceCharacteristic*, the **Standard Description** of any *Requirement* that **specifies** the current *Function* that **motivates** the *PerformanceCharacteristic* is also printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is **described by** a *Section* in the document.

>PRINT FUNCTION, RQMTS AND GRAPH

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section*. If that element is a *Function* and the previous *Section* has no **describes** relationship to a *PerformanceCharacteristic,* the **Standard Description** of every *Requirement* that **specifies** that *Function* is printed using the **Requirement Formatting Rules** If the previous section describes a *PerformanceCharacteristic*, the **Standard Description** of any *Requirement* that **specifies** the current *Function* that **motivates** the *PerformanceCharacteristic* is also printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is

described by a *Section* in the document.

This *ReportBlock* may also print a graph based on the *FNet*, *RNet*, or *TNet* (if it is not empty) that is the **current decomposition of** the *Function*. The graphs available are a Behavior Diagram, or FFBD, or IDEF0, or an FFBD and IDEF0. The default is to not print any graph. If a graph is to be printed, the type of graph must be selected from the report's run-time options.

>PRINT FUNCTION, RQMTS, AND I/O)

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section.* If that element is a *Function* and the previous *Section* has no **describes** relationship to a *PerformanceCharacteristic,* the **Standard Description** of every *Requirement* that **specifies** that *Function* is printed using the **RequirementFormattingRules** If the previous section describes a *PerformanceCharacteristic*, the **Standard Description** of any *Requirement* that **specifies** the current *Function* that **motivates** the *PerformanceCharacteristic* is also printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is **described by** a *Section* in the document.

The *ReportBlock* also prints an Input/Output Table identifying all *TimeItems*/*DiscreteItems* **input to/output from** the *Function*. The table consists of the following:

- Column One: The **Standard Title** of an *Item* that is **input to**/**output from** the *Function*.
- Column Two: The Description of the *Item* listed in Column One.

>PRINT FUNCTION, RQMTS, I/O, AND GRAPH

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section.* If that element is a *Function* and the previous *Section* has no **describes** relationship to a *PerformanceCharacteristic,* the **Standard Description** of every *Requirement* that **specifies** that *Function* is printed using the **RequirementFormattingRules** If the previous section describes a *PerformanceCharacteristic*, the **Standard Description** of any *Requirement* that **specifies** the current *Function* that **motivates** the *PerformanceCharacteristic* is also printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is **described by** a *Section* in the document.

The *ReportBlock* also prints an Input/Output Table identifying all *TimeItems*/*DiscreteItems* **input to/output from** the *Function*. The table consists of the following:

- Column One: The **Standard Title** of an *Item* that is **input to**/**output from** the *Function*.
- Column Two: The Description of the *Item* listed in Column One.

This *ReportBlock* may also print a graph based on the *FNet*, *RNet*, or *TNet* (if it is not empty) that is the **current decomposition of** the *Function*. The graphs available are a Behavior Diagram, or FFBD, or IDEF0, or an FFBD and IDEF0. The default is to not print any graph. If a graph is to be printed, the type of graph must be selected from the report’s run-time options.

>PRINT FUNCTION, RQMTS, I/O, GRAPH AND GRAPH REFERENCES

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section.* If that element is a *Function* and the previous *Section* has no **describes** relationship to a *PerformanceCharacteristic,* the **Standard Description** of every *Requirement* that **specifies** that *Function* is printed using the **Requirement Formatting Rules** If the previous section describes a *PerformanceCharacteristic*, the **Standard Description** of any *Requirement* that **specifies** the current *Function* that **motivates** the *PerformanceCharacteristic* is also printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is **described by** a *Section* in the document.

The *ReportBlock* also prints an Input/Output Table identifying all *TimeItems*/*DiscreteItems* **input to/output from** the *Function*. The table consists of the following:

- Column One: The **Standard Title** of an *Item* that is **input to**/**output from** the *Function*.
- Column Two: The Description of the *Item* listed in Column One.

This *ReportBlock* may also print a graph based on the *FNet*, *RNet*, or *TNet* (if it is not empty) that is the **current decomposition of** the *Function*. The graphs available are a Behavior Diagram, or FFBD, or IDEF0, or an FFBD and IDEF0. The default is to not print any graph. If a graph is to be printed, the type of graph must be selected from the report’s run-time options. Note: IDEF0 and DFD are not valid graph choices for *DiscreteFunctions*. Only a Behavior Diagram can be used for *Transforms*.

If the *Function* is a *DiscreteFunction/Transform* **decomposed by** an *RNet/TNet* with a non-null decomposition and the report’s run-time options to print graphs was selected, a table clarifying the graph nodes is printed before the graph. The table consists of the following:

- Column One: The element type and **Standard Title** of an element **referred by** the *RNet*/*TNet* if that element has a non-empty Description.
- Column Two: The Description of the element identified in Column One. If the element is a *Condition*, one of the following is printed if it is available:
  - The *Condition* Expression if it is not empty.
  - The name of the *ProbabilityDistribution* that **supplies value to** the *Condition*.
  - The *Condition* Exit Probability if it is not empty.

>PRINT INTERFACES W REFERENCE

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of any elements **described by** the *Section.* If a **described by** element is an *Interface*, an **Interface document reference** is printed as a separate paragraph if the reference exists. An **Interface document reference** is either:

- A *ReportDocument* with Document Type `IRS` that **contains** a *Section* that **describes** the *Interface*, or
- A *Source* with Source Type `IRS`, `IDD`, or `ICD` that **documents** the *Interface*.

Only one reference is printed. The precedence is *ReportDocument*, *Source* of Source Type `IRS`, *Source* of Source Type `IDD`, *Source* of Source Type `ICD`. The reference consists of the element Title or name and the Document Control Number.

>PRINT INTERNAL INTERFACE IDENTIFICATION

This *ReportBlock* prints the **Standard Section Titling** and a table listing the internal *Interfaces* for the *Component* **reported on** by the current document.

The table produced by this *ReportBlock* contains the following:

- Column One: The **Standard Title** of each *Interface* that is **connected thru** a *Component* in the **Build From List** and **connects to** *Components* only on the **Build From List** and/or descendents of the **Build From List** elements.
- Column Two: The **Standard Title** of each *Component* on the **Build From List** that **connects thru** the *Interface* in Column One.

>PRINT INTERNAL ITEM USAGE TABLE

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section,* and, if that element is a *TimeItem/DiscreteItem*, a two column table showing the capability usage of that *Item*.

The following term needs to be defined:

- A **Leaf Capability** is either a *DiscreteFunction* **described by** a *Section* in the current Document with Number = `3.2.*` or a *TimeFunction* **described by** a leaf-level *Section* in the current Document with Number = `3.2.*`.

The table for this *ReportBlock* contains:

- Column One: Every **Leaf Capability** that **outputs** the *TimeItem/DiscreteItem* **described by** the current *Section.*
- Column Two: Every **Leaf Capability** that **inputs** the *TimeItem/DiscreteItem* **described by** the current *Section*.

>PRINT IRS INTERFACE DIAGRAMS

This *ReportBlock* prints the **Standard Section Titling** and an interface diagram for each *Component* in the **Reported List**. An interface diagram consists of two boxes of *Components* with a list of *ItemLinks* between the boxes.

The left hand box of the diagram contains the Title or name of the current *Component* from the **Reported List.**

The right hand box of the diagram contains, within sub-boxes, the Titles or names of the **Interfacing Components** An **Interfacing Component** is the highest level *Component* in the component hierarchy which is an ancestor of the *Component* which **inputs from**/**outputs to** an *ItemLink* which **exits**/**enters** the current *Component* and which also is **entered by**/ **exited by** the *ItemLink*. If there is an **Include List**, the **Interfacing Component** is further restricted to be a *Component* on the **Include List** or the **Reported List**. Any *Component* with Component Type = `Environment` is ignored.

In the space between the two boxes is listed the Title or name of every *ItemLink* that interfaces between the current *Component* and each **Interfacing Component** An arrow (<-- or -->) is used to show the direction of the flow.

>PRINT IRS INTERFACE IDENTIFICATION

This *ReportBlock* prints the **Standard Section Titling** and a three column table listing *Interfaces*, *ItemLinks* and their associated source and destination *Components*.

An *ItemLink* and its **contained by** *Interface* are included in the table if it **exits**/**enters** a *Component* **reported by** the current *ReportDocument* and, if the *ReportDocument* **includes** *Components*, **enters**/**exits** a *Component* that **is included in** the *ReportDocument*.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the *Interface*.
- Column Two: The **Standard Title** of each *ItemLink* that is **contained by** the *Interface* and meets the criteria identified above.
- Column Three: The **Standard Title** for the *Components* which are the interface level source and destination for the *ItemLink* in Column Two.

A one sentence lead-in identifying the *Interfaces* covered in the specification is automatically printed before the table. For each *Interface*, the *Interface* **Standard Title** is printed.

>PRINT IRS INTERFACE, RQMTS, ITEMLINK TABLE

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of any elements **described by** the *Section.* If the **described by** element is an *Interface*, the Standard Description of every *Requirement* that **specifies** that element is printed using the **Requirement Formatting Rules**

If the **describes** element is an *Interface*, a two column table is also printed listing the *ItemLinks* **contained by** the *Interface* and their associated source and destination *Components*. An *ItemLink* is included in the table if it **exits**/**enters** a *Component* **reported by** the current *ReportDocument* and, if the *ReportDocument* **includes** *Components*, **enters**/**exits** a *Component* that **is included in** the *ReportDocument*.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of an *ItemLink*.
- Column Two: The **Standard Title** for the *Components* which are the interface level source and destination for the *ItemLink* in Column One.

A one sentence lead-in is automatically printed before the table.

>PRINT IRS ITEMLINK MESSAGE TABLE

This *ReportBlock* prints the **Standard Section Titling**, and, if the **describes** element is an *ItemLink*, a table listing the *TimeItems*/*DiscreteItems* **carried by** the ItemLink.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the *DiscreteItem*/*TimeItem*.
- Column Two: The Description of the *DiscreteItem*/*TimeItem* listed in Column One.

A one sentence lead-in is automatically printed before the table. This sentence lists the source and destination *Components* for the *ItemLink*.

>PRINT ITEM DEFINITION

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of any elements **described by** the *Section.* If the **described by** element is a *TimeItem*/*DiscreteItem*, the characteristics of the members of the *Item* decomposition hierarchy are printed in tabular form. For *TimeItems* these members may be *TimeItems* or *DiscreteItems*; for *DiscreteItems* these members may be *AbstractObjectTypes* or *AttributeTypes*. A *DiscreteItem* **contains** *AbstractObjectTypes*; an *AbstractObjectType* **consists of** other *AbstractObjectTypes* and/or is **defined by** *AttributeTypes*.

The table for this *Section* contains:

- Column One: The name of the data element and its Description attribute if the element is not the **described by** *TimeItem*/*DiscreteItem*.
- Column Two:
    - If the data element is a *TimeItem*, it is treated as a Data Collection and the *Items* in its decomposition are listed by name.
    - If the data element is a *DiscreteItem* and it has the **contains** relationship, the targets of the **contains** relationship are listed.
    - If the data element is a *DiscreteItem* and it does not have the **contains** relationship, its characteristics are listed: Units, Range, Accuracy, Precision/Resolution and Data Type.
    - If the data element is an *AbstractObjectType*, the targets of the **consists of** and **defined by** relationships are listed.
    - If the data element is an *AttributeType*, its characteristics are listed: Units, range as a composite of MinimumValue and MaximumValue, Accuracy, Precision/Resolution and Data Type.

>PRINT ITEM DEFINITION, RQMTS

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of any elements **described by** the *Section.* If the **described by** element is a *TimeItem*/*DiscreteItem*, the **Standard Description** of every *Requirement* that **specifies** the element is printed using the **Requirement Formatting Rules**

If the **described by** element is a *TimeItem*/*DiscreteItem*, the characteristics of the members of the *Item* decomposition hierarchy are printed in tabular form. For *TimeItems* these members may be *TimeItems* or *DiscreteItems*; for *DiscreteItems* these members may be *AbstractObjectTypes* or *AttributeTypes*. A *DiscreteItem* **contains** *AbstractObjectTypes*; an *AbstractObjectType* **consists of** other *AbstractObjectTypes* and/or is **defined by** *Attribute-*

Types.

The table for this *Section* contains:

- Column One: The name of the data element and its Description attribute if the element is not the **described by** *TimeItem*/*DiscreteItem*.
- Column Two:
  - If the data element is a *TimeItem*, it is treated as a Data Collection and the *Items* in its decomposition are listed by name.
  - If the data element is a *DiscreteItem* and it has the **contains** relationship, the targets of the **contains** relationship are listed.
  - If the data element is a *DiscreteItem* and it does not have the **contains** relationship, its characteristics are listed: Units, Range, Accuracy, Precision/Resolution and Data Type.
  - If the data element is an *AbstractObjectType*, the targets of the **consists of** and **defined by** relationships are listed.
  - If the data element is an *AttributeType*, its characteristics are listed: Units, range as a composite of MinimumValue and MaximumValue, Accuracy, Precision/Resolution and Data Type.

>PRINT MAJOR COMPONENT FUNCTIONAL INTERFACES

This *Section* prints the **Standard Section Titling** and two tables based on the *Functions* listed in the Performance Characteristics Table in Section 3.7.x.1.

The first table lists receiving *Functions* from the Performance Characteristics Table that are **allocated to** the current *Component* identified in Section 3.7.x and **input** *TimeItems/DiscreteItems* that are also **output from** *Functions* **allocated to** other *Components* in the **Build From List**.

The table for this *ReportBlock* contains:

- Column One: The **Standard Title** of a Performance Characteristics Table *Function* which **inputs** a *TimeItem/DiscreteItem* that originates in a *Function* **allocated to** the current *Component*.
- Column Two: The **Standard Title** of the *TimeItems/DiscreteItems* that qualified the Column One *Function* as a receiving capability.
- Column Three: The **Standard Title** of any *Component* and the sending *Functions* **allocated to** it, where a *Function* **outputs** a *TimeItem/DiscreteItem* from Column Two.

The second table lists sending *Functions* from the Performance Characteristics Table that are **allocated to** the current *Component* and **output** *TimeItems/DiscreteItems* that are also **input to** *Functions* **allocated to** other *Components* in the **Build From List**.

The table for this *ReportBlock* contains:

- Column One: The **Standard Title** of a Performance Characteristics Table *Function* which **outputs** a *TimeItem/DiscreteItem* that is sent to a *Function* **allocated to** the current *Component***.**

- Column Two: The **Standard Title** of the *TimeItems/DiscreteItems* that qualified the Column One *Function* as a sending capability.
- Column Three: The **Standard Title** of any *Component* and the receiving *Functions* **allocated to** it, where a *Function* **inputs** a *TimeItem/DiscreteItem* from Column Two.

>PRINT MAJOR COMPONENT LIST

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Titles** of the *Components* in the **Build From List**.

>PRINT MAJOR COMPONENT PERFORMANCE CHARACTERISTICS

This ReportBlock prints the **Standard Section Titling** and, if the *Section* **describes** a *Component*, a table listing *Sections* and *Requirements* associated with *Functions* **allocated to** the reference *Component* and any **Stand-alone** *Requirement* that **specifies** the referenced *Component*. A *Requirement* is **Stand-alone** if it **specifies** a *PerformanceCharacteristic* (that is **described by** a *Section* that **is contained by** the current *ReportDocument*) and does not have the **specifies** relationship to a *Function* that **motivates** the *PerformanceCharacteristic.*

The table produced by this *ReportBlock* contains:

- Column One: The Number of any *Section* in the current document (with a Number = `3.2.1.*`) that either
  - **describes** a *Function* that is **allocated to** the *Component* **described by** the current *Section*,
  - **describes** a *PerformanceCharacteristic* that is **motivated by** a *Function* that is **allocated to** the *Component* **described by** the current *Section*, or
  - **describes** a *PerformanceCharacteristic* that **specifies** a **Stand-alone** *Requirement* that **specifies** the *Component.*
- Column Two: The Title of the *Section* identified in Column One.
- Column Three: The Project Unique ID of:
  - each *Function* associated with the *Section* referenced in Column One that has a “shall” statement in its Description and is **allocated to** the reference *Component* and of every *Requirement* printed in the *Section* referenced in Column One or printed in a numbered subsection below the *Section* referenced in Column One, and
  - each **Stand-alone** *Requirement* that **specifies** the *Component* and the *PerformanceCharacteristic* **described by** the *Section* referenced in Column One.

>PRINT MAJOR COMPONENT PHYSICAL CHARACTERISTICS

This *Section* prints the **Standard Section Titling** and a table listing every *Requirement* that

specifies the *Component* **described by** the current parent *Section* and is **described by** a *Section* with a Number > `3.2.2` and < `3.7` that **is contained by** the current *ReportDocument*.

The table for this *ReportBlock* contains:

- Column One: The Number of the *Section* (with a Number > `3.2.2` and < `3.2.7`) that **describes** a *Requirement* that **specifies** the *Component* **described by** the current *Section*.
- Column Two: The Title of the *Section* identified in Column One.
- Column Two: The Project Unique ID of every *Requirement* that is **described by** the *Section* listed in Column One.

>PRINT PCS

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section*, and, if that element is a *PerformanceCharacteristic*, the **Standard Description** of its **specified by Stand-alone** *Requirements* are printed using the **Requirement Formatting Rules**. A *Requirement* is **Stand-alone** if it **specifies** the current *PerformanceCharacteristic* and does not have the **specifies** relationship to a *Function* that **motivates** the *PerformanceCharacteristic.*

For a *PerformanceCharacteristic* of Type `Capability`, the **Standard Description** of any *Requirement* that **specifies** a *Function* that **motivates** the *PerformanceCharacteristic* and is not **described by** a Section in the document is also printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is **described by** a *Section* in the document.

>PRINT PCS, FCNS

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section*, and, if that element is a *PerformanceCharacteristic*, the **Standard Description** of its **specified by Stand-alone** *Requirements* are printed using the **Requirement Formatting Rules**. A *Requirement* is **Stand-alone** if it **specifies** the current *PerformanceCharacteristic* and does not have the **specifies** relationship to a *Function* that **motivates** the *PerformanceCharacteristic.*

For a *PerformanceCharacteristic* of Type `Capability`, the **Standard Description** of any *Function* that **motivates** the *PerformanceCharacteristic* is also printed if the *Function* is not **described by** a *Section* in the document. The **Standard Description** of any *Requirement* that **specifies** the *Function* is then printed (using the **Requirement Formatting Rules**) unless either that *Requirement* **specifies** another *PerformanceCharacteristic* **described by** a *Section* in the document and does not also specify the current *PerformanceCharacteristic* or the *Requirement* is **described by** a *Section* in the document.

>PRINT PREVIOUSLY REFERENCED PARAGRAPH

This *ReportBlock* is used by the Output Sections Report to provide backward referencing when the user has selected the report's run-time options to reference *Sections* previously presented in this *ReportDocument* rather than print the information again.

This *ReportBlock* is substituted automatically for the *ReportBlock* referenced in the Report Block name of the current *Section* only if the *Section* **describes** just one non-*TextBlock* element that is also **described by** another *Section* which also **is contained in** the current *ReportDocument*. The other *Section* must have a lower Number than the current *Section* (i.e., it appears earlier in the document) and must have the same Report Block name value as the current *Section*.

If all of these criteria are met, this *ReportBlock* will be executed and it will refer to the Number of the first *Section* which **describes** the element.

>PRINT REPORTED ON ELEMENT'S COMPONENT HIERARCHY

This *ReportBlock* prints the **Standard Section Titling** and a Component Hierarchy of the *Component* **reported by** the *ReportDocument.*

>PRINT REPORTED ON ELEMENT'S COMPONENT VIEW

This *ReportBlock* prints the **Standard Section Titling** and a Component View of the *Component* **reported by** the *ReportDocument.*

>PRINT REPORTED ON ELEMENT'S CONTEXT DIAGRAM

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of any elements **described by** the *Section*. If the *Component* **reported by** the *ReportDocument* **has context** of an *FNet*, the Behavior Diagram of the *FNet* is printed.

>PRINT REPORTED ON ELEMENT'S DESCRIPTION

This *ReportBlock* prints the **Standard Section Titling** and the **Standard Description** of the *Component* **reported by** the *ReportDocument*.

>PRINT REPORTED ON ELEMENT'S DESCRIPTION & CONTEXT DIAGRAM

This *ReportBlock* prints the **Standard Section Titling** and, if the *Component* **reported by** the *ReportDocument* **has context** of an *FNet*, it prints the *Component* Description and the Behavior Diagram of the *FNet*.

>PRINT REPORTED ON ELEMENT'S TOP-LEVEL BEHAVIOR

This *ReportBlock* prints the **Standard Section Titling** and, if the *Component* **reported by** the *ReportDocument* **performs** a *TimeFunction*, the *TimeFunction's* Behavior Diagram is printed.

>PRINT REQUIREMENTS SUBPARAGRAPHS ONLY

This *ReportBlock* prints the **Standard Section Titling** and, for each *Requirement* that **specifies** an element **described by** the *Section*, the **Standard Description** of the *Requirement* following the **Requirement Formatting Rules**

>PRINT REQUIREMENTS TRACEABILITY MATRIX

This *ReportBlock* prints the **Standard Section Titling** and a Requirements Traceability Matrix. For each Requirement in the current document, the matrix provides the traceability back to a parent specification *Requirement* or to a documenting *Source*.

Because of the complexity of the search for a documented element in this *ReportBlock*, these terms need to be defined:

- The **Parent Document List** consists of any *ReportDocuments* which the current *ReportDocument* is **traced from.**
- A **Valid Internal Source** is a *ReportDocument* on the **Parent Document List.**
- A **Valid External Source** is a *Source* that has a Source Type = `Originating Requirements` or `Change Notice` or `Statement of Work`.
- An **Externally Documented element** is a *Requirement* which is **documented by** a *Source* that is a **Valid External Source.**
- An **Internally Documented element** is either a *Requirement/Function* that is **described by** a *Section* that is **contained by** a *ReportDocument* that is a **Valid Internal Source** or is a *Requirement* that either:
  - **specifies** a *TimeFunction* / *DiscreteFunction* that is **described by** a *Section* that **is contained by** a *ReportDocument* that is a **Valid Internal Source,**
  - **specifies** a *TimeFunction* / *DiscreteFunction* that is **motivated by** a *PerformanceCharacteristic* which is **described by** a *Section* that both **is contained by** a *ReportDocument* that is a **Valid Internal Source** and prints the *Requirement.* (See `>Print PCs` for the rules by which a *Requirement* is printed in a *Section* which **describes** a *PerformanceCharacteristic*.), or
  - **specifies a** *PerformanceCharacteristic* which is **described by** a *Section* that **is contained by** a *ReportDocument* that is a **Valid Internal Source.**
- An **Initial Requirement** is one of the following:
  - a *Requirement* that either is **described by** a *Section* in the **Traceability List, specifies** a *TimeFunction* / *DiscreteFunction* that is **described by** a *Section* in the **Traceability List, specifies** a *TimeFunction* / *DiscreteFunction* that is **motivated by** a *PerformanceCharacteristic* that is **described by** a *Section* that both is in the **Traceability List** and prints the *Requirement.* (See `>Print PCs` for the rules by which a *Requirement* is printed in a *Section* which **describes** a *PerformanceCharacteristic*.), or **specifies** a *PerformanceCharacteristic* that is **described by** a *Section* in the **Traceability List.**
  - a *DiscreteFunction* / *TimeFunction* / *Transform* that contains a “shall” statement and is **described by** a *Section* in the **Traceability List**.

There are three search algorithms: **Search for a Valid External Source**, **Search for a Valid Internal Source**, and **Search for a traced from target.** These algorithms are used to find an associated **Documented element.** These algorithms are combined in the Initial Search for a documented element and the Continuing Search for a documented element

- **Search for a Valid External Source**: If the current element is a *Requirement* that is an **Externally Documented element**, the current element and its associated *Source* which is a **Valid External Source** are added to the list of documented elements and the search against the current element ends. If the current element is a *Function* that is **justified by** a *Requirement* that is an **Externally Documented element**, the *Requirement* and its *Source* are added to the list of documented elements and the search against the current element ends.

- **Search for a Valid Internal Source**: If the current element is a *Requirement* that is an **Internally Documented element**, the current element and each *Section-ReportDocument* pair which is a **Valid Internal Source** are added to the list of documented elements and the search ends on the current element. If the current element is a *Function* that is **justified by** a *Requirement* that is an **Internally Documented element**, the *Requirement* and each *Section-ReportDocument* pair which is a **Valid Internal Source** are added to the list of documented elements and the search ends on the current element.
- **Search for a traced from target**: For the current element, each target of the **traced from** relationship is examined. If the target is a *CriticalIssue* / *Decision*, the **Search for a traced from target** is done for the *CriticalIssue* /*Decision*. If that target is a *Function* or *Requirement* and it is not a **Documented element**, the target becomes the current element and the **Continuing Search** is executed.

Initial Search for a documented element

With an **Initial Requirement** as the current element, the following searches are done. Each search unit is executed only if the preceding step did not find at least one **Documented element**.

- If the **ParentDocumentList** is not empty, the **SearchforValidInternalSource** is executed for the **Initial Requirement**. Otherwise, the **Search for Valid External Source** is executed for the **Initial Requirement**.
- The **Continuing Search** (described below) is executed for the **Initial Requirement**.

Continuing Search for a documented element

With any current element, the following searches are done.

- A Search for a parent element is executed for the current element.

```
    Requirement (current element)
        incorporated by
            Requirement (parent)
Or
    DiscreteFunction/TimeFunction/Transform (current element)
        referred by
            FNet/Rnet/TNet
                current decomposition of
                    DiscreteFunction/TimeFunction/Transform
```

For each parent the searches listed below are done. Each search is executed only if the preceding search did not find at least one **Documented element**.

  - If the **Parent Document List** is not empty, the **Search for Valid Internal Source** is executed for the parent element. Otherwise, the **Search for Valid External Source** is executed for the parent element.

  - The parent *Requirement* becomes the current element and the Continuing Search is executed for it.

- The **Search for a traced from target** is executed for the current element.

For each *Section* in the **Traceability List**, the Requirements Traceability Matrix lists:

- Column One: *Section* Number
- Column Two: *Section* Title
- Column Three: Project Unique ID for each “shall” statement *Function* and *Requirement* printed in the document section referenced in Columns One and Two. If there is a *Function*/*Requirement*, but it’s Project Unique ID is undefined, this column contains ‘--’. If there are no *Requirements* printed in the *Section*, this column contains ‘N/A’ and all following columns are empty.
- Column Four: For each *Function*/*Requirement* listed in Column Three, the Project Unique ID of each **Documented element** *Function*/*Requirement*. If there is a **Documented element**, but it’s Project Unique ID is undefined, this column contains ‘--’. If the **Documented element** is the same element as the element identified in Column Three, this column is empty. If there are no **Documented elements**, this column and all following columns are empty.
- Column Five: Paragraph Number of the **Documented element** identified in Column Four for the document identified in Column Seven.
- Column Six: Paragraph Title of the **Documented element** identified in Column Four for the document identified in Column Seven.
- Column Seven: Document Control Number for the *ReportDocument* or *Source* that documents the **Documented element**. If there is no Document Control Number, the Title or Name of the document is listed.

>PRINT REQUIREMENTS VERIFICATION MATRIX

This *ReportBlock* prints the **Standard Section Titling** and a Requirements Cross Reference Matrix. An entry is made in the table for each *Section* in the **Traceability List**.

```
Section (Traceability List)
    describes
        Requirement
    OR
    describes
        TimeFunction / DiscreteFunction /Transform
            specified by
                Requirement
    OR
    describes
        PerformanceCharacteristic
            motivated by
                TimeFunction / DiscreteFunction
                    specified by
                        Requirement
            specified by
                Requirement
```

```
TimeFunction / DiscreteFunction /Transform / Requirement
    has verification method of
        VerificationMethod
    OR
    verified by
        VerificationRequirement
            has verification method of
                VerificationMethod
                    described by
                        Section
                            is contained in
                                ReportDocument
```

The Requirements Cross Reference Matrix contains:

- Column One: The Number of a *Section* from the **Traceability List**.
- Column Two: The Title of that *Section*.
- Column Three: The Project Unique ID for every "shall" statement *Function* and *Requirement* printed in the *Section*. If there is a *Function*/*Requirement*, but it's Project Unique ID is undefined, this column contains '--'. If there are no *Functions*/*Requirements* printed in the *Section*, this column contains 'N/A'.
- Column Four: The Method of a *VerificationMethod* which is either the **verification method for** the *Function*/*Requirement* identified in that row of the table or the **verification method for** a *VerificationRequirement* that **verifies** the *Function*/*Requirement*.
- Column Five: The Level of the *VerificationMethod* whose Method is listed in Column Four.
- Column Six: The Number of every *Section* that **is contained in** the current *ReportDocument* and **describes** a *VerificationRequirement* that **verifies** a *Function*/*Requirement* listed in Column Three.

>PRINT SRS INTERNAL DATA TABLE

This *ReportBlock* prints the **Standard Section Titling** and a three column table containing CSCI internal data elements. Internal data elements are *TimeItems*/*DiscreteItems* which are both **input to** and **output from Leaf Capability Functions**. A **Leaf Capability Function** is either a *TimeFunction* **described by** a leaf-level *Section* with a Number = `3.2.*` that **is contained by** the current *ReportDocument* or a *DiscreteFunction* **described by** a *Section* with a Number = `3.2.*` that **is contained by** the current *ReportDocument*

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the data element.
- Column Two: The **Standard Title** of each **Leaf Capability Function** that **outputs** the *Item* listed in Column One.
- Column Three: The **Standard Title** of each **Leaf Capability Function** that **inputs** the *Item* listed in Column One.

>PRINT SRS EXTERNAL DATA TABLE

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of any elements **described by** the *Section* and a table listing the external interface data elements for the *Component* **reported by** the *ReportDocument*. The external interface data elements are those *TimeItems/DiscreteItems* **carried by** an *ItemLink* which either **enters** or **exits** the *Component* **reported by** the *ReportDocument*. An *ItemLink* which is linked to a *Component* with Component Type = `Environment` is ignored.

The table for this *ReportBlock* contains:

- Column One: The **Standard Title** of the *ItemLink*, the direction of the link, the interfacing *Component* and an **Interface document reference** if the reference exists. An **Interface document reference** is either:
    - A *ReportDocument* with Document Type `IRS` that **contains** a *Section* that **describes** the *ItemLink*, or
    - A *Source* with Source Type `IRS`, `IDD`, or `ICD` that **documents** the *ItemLink*.

  Only one reference is printed. The precedence is *ReportDocument*, *Source* of Source Type `IRS`, *Source* of Source Type `IDD`, *Source* of Source Type `ICD`. The reference consists of the element Title or name and the Document Control Number.
- Column Two: The **Standard Title** of each *TimeItem*/*DiscreteItem* carried by the *ItemLink* listed in Column One.
- Column Three: The **Standard Title** of each **Leaf Capability Function** which **inputs**/**outputs** the *Item* in Column Two. A **Leaf Capability Function** is either a *TimeFunction* **described by** a leaf-level Section with Number = `3.2.*` that **is contained by** the *ReportDocument* or a *DiscreteFunction* **described by** a *Section* with Number = `3.2.*` that **is contained by** the *ReportDocument*.

>PRINT SSDD CI ALLOCATED REQUIREMENTS

This *Section* prints the **Standard Section Titling** and, if the *Section* describes a *Component*, the Allocated Requirements table for that *Component*. The table identifies all *Functions* **allocated to** the *Component*, their **specifies** *Requirements*, and the *Source/ReportDocument* which is the specification source for the *Requirements*. The table also identifies all **Stand-alone** *Requirements* allocated to the *Component* and their specification source. A *Requirement* is **Stand-alone** if it **specifies** a *PerformanceCharacteristic* and does not have the **specifies** relationship to a *Function* that **motivates** the *PerformanceCharacteristic.* This *ReportBlock* uses the same traceability search as >Print Requirements Traceability Matrix.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of a *TimeFunction/DiscreteFunction* **allocated to** the current *Component* or of a *PeformanceCharacteristic* that has a least one **Stand-alone** *Requirement* that **specifies** the current *Component.*
- Column Two: The Project Unique ID of each *Requirement* which either **specifies** the *Function* in Column One or is a **Stand-alone** *Requirement* of the *PerformanceCharacteristic* in Column One and **specifies** the current *Component*.

- Column Three: The Paragraph Number and Paragraph Title of the source *Requirement* (if the *Requirement* is **documented by** a **Valid External Source**) or the source *Section* Number and *Title* (if the **documented element** is associated with a **Valid Internal Source**) and the Document Control Number of the *Source*/*ReportDocument* that is the documentation source of the *Requirement* in Column Two.

>PRINT SSDD CI DESIGN CONSTRAINTS

This *Section* prints the **Standard Section Titling** and, if the *Section* describes a *Component*, the Design Constraints table for that *Component*. The table identifies each non-functional *Requirements* that **specifies** the *Component,* and the *Source/ReportDocument* which is the specification source for the *Requirements*. This *ReportBlock* uses the same traceability search as >Print Requirements Traceability Matrix.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of each *Requirement* that **specifies** the current *Component.*
- Column Two: The Paragraph Number and Paragraph Title of the source *Requirement* (if the *Requirement* is **documented by** a **Valid External Source**) or the source *Section* Number and *Title* (if the **documented element** is associated with a **Valid Internal Source**) and the Document Control Number of the *Source*/*ReportDocument* that is the documentation source of the *Requirement* in Column One.

>PRINT SSDD CI SYSTEM EXTERNAL INTERFACES

This *Section* prints the **Standard Section Titling,** the **Standard Description** of any elements **described by** the *Section* and, if the element is an *ItemLink*, a table listing the *TimeItems*/*DiscreteItems* **carried by** the *ItemLink*.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the *DiscreteItem*/*TimeItem*.
- Column Two: The Description of the *DiscreteItem*/*TimeItem* listed in Column One.

A one sentence lead-in is automatically printed before the table. This sentence lists the source and destination *Components* for the *ItemLink*.

>PRINT SSDD CI UTILIZATION TABLE

This *ReportBlock* prints the **Standard Section Titling,** the **Standard Description** of each element **described by** the *Section*, and, if that element is a *Component*, a three column table listing the CSCIs that utilize the *Component* and the extent of their utilization.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of a *Component* with ComponentType `CSCI` that **executes on** the *Component* **described by** the current *Section*.
- Column Two: The SW, Percent of Memory Utilization attribute of the *Component* listed in Column One.
- Column Three: The SW, Percent of Processor Utilization attribute of the *Compo-*

nent listed in Column One.

A one sentence lead-in is automatically printed before the table.

>PRINT SSDD INTERNAL INTERFACES

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section*, and, if that element is an *ItemLink*, a two column table listing the *DiscreteItems*/*TimeItems* **carried by** the *ItemLink*.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the *DiscreteItem*/*TimeItem*.
- Column Two: The Description of the *DiscreteItem*/*TimeItem* listed in Column One.

A one sentence lead-in is automatically printed before the table. This sentence lists the source and destination **Build From List** *Components* for the *ItemLink*.

>PRINT SSDD REQUIREMENTS TRACEABILITY MATRIX

This *ReportBlock* prints the **Standard Section Titling** and a three column table containing the traceability for the *Requirements* cited in the current document. This *ReportBlock* uses the same traceability search as >Print Requirements Traceability Matrix.

The table produced by this *ReportBlock* contains:

- Column One: The Project Unique ID of every *Requirement* that either **specifies** a *Component* on the **Build From List** or **specifies** a *TimeFunction*/*DiscreteFunction* **allocated to** a *Component* on the **Build From List**. If the *Requirement* **specifies** a *TimeFunction*/*DiscreteFunction* the **Standard Title** of the *Function* is also printed.
- Column Two: The Paragraph Number and Paragraph Title of the source *Requirement* (if the *Requirement* is **documented by** a **Valid External Source**) or the source *Section* Number and *Title* (if the **document element** is associated with a **Valid Internal Source**) and the Document Control Number of the *Source*/*ReportDocument* that is the documentation source of the *Requirement* in Column One.
- Column Three: The **Standard Title** of all *Components* on the **Build From List** that are associated with the *Requirement* listed in Column One.

>PRINT SSDD SYSTEM EXTERNAL INTERFACES

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section*, and, if that element is an *Interface*, a two column table of *ItemLinks* and *Components*, showing the interfaces between subordinate elements of the *Component* **reported by** the *ReportDocument* and all external *Components*. An *ItemLink* is included in the table if it both **is contained by** the *Interface* and **exits** or **enters** the *Component* **reported by** the *ReportDocument*.

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the *ItemLink*.
- Column Two: The **Standard Title** of the *Component* on the **Build From List** which the *ItemLink* **exits**/**enters** and the **Standard Title** of the highest level *Component* in the

component hierarchy which is an ancestor of the *Component* which **inputs from**/**outputs to** the *ItemLink* and also is **entered by**/**exited by** the *ItemLink*.

>PRINT SSDD SYSTEM INTERNAL INTERFACES

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of each element **described by** the *Section*, and, if that element is an *Interface*, a two column table of *ItemLinks* and *Components*, showing the interfaces between subordinate elements of the *Component* **reported by** the *ReportDocument* (i.e., the elements on the **Build From List**).

The table produced by this *ReportBlock* contains:

- Column One: The **Standard Title** of the *ItemLink*.
- Column Two: The **StandaTitle** of the *Components* on the **BuildFromList** which are **entered by**/**exited by** the *ItemLink* in Column One.

>PRINT STATE, MODE, CAPABILITY/FCN TABLE

This *ReportBlock* prints the **Standard Section Titling** and, if there are *PerformanceCharacteristics* of Type `State` and/or `Mode` for the *Component* reported by the current *ReportDocument*, a table listing by State and/or Mode the *PerformanceCharactertistics* of Type `Capability` or, if there are no *PerformanceCharacteristics* of Type `Capability`, any *TimeFunction*/*DiscreteFunction* that **motivates** a lowest-level *PerformanceCharacteristic*.

In order to determine the table contents, the following element relationships are followed:

```
ReportDocument
    reports on
        Component
            exhibits
                PerformanceCharacteristic
                    includes
                        PerformanceCharacteristic
                            motivated by
                                TimeFunction/DiscreteFunction
```

If there are no states or modes, a statement to that affect is automatically printed.

>PRINT STATE, MODE, CI TABLE

This *ReportBlock* prints the **Standard Section Titling** and, if there are *PerformanceCharacteristics* of Type `State` and/or `Mode` for the *Component* **reported by** the current *ReportDocument*, a table is produced that correlates the subcomponents to the States and/or Modes. A *Component* correlates to a State and/or Mode if the *Component* is in the **Build From List** and **allocates** a *TimeFunction*/*DiscreteFunction* that **motivates** a leaf -level *PerformanceCharacteristic* in the particular State/Mode hierarchy.

In order to determine the table contents, the following element relationships are followed:

```
ReportDocument
    reports on
        Component
            exhibits
```

```
*PerformanceCharacteristic*
    **includes**
        *PerformanceCharacteristic*
            **motivated by**
                *TimeFunction/DiscreteFunction*
                    **allocated to**
                        *Component (in* **Build From List***).*
```

>PRINT SUBORDINATE ELEMENTS RQMTS TABLE

This *ReportBlock* prints the **Standard Section Titling**, the **Standard Description** of any elements **described by** the *Section* and, if the element is a *Component*, a table showing the *Functions* (capabilities)*/Requirements* that are directly/indirectly allocated to the *Component* **described by** the *Section*. The table contains the following:

- ❑ Column One: The Number of any *Section* that **is contained by** the current *ReportDocument* and that either
  - ■ **describes** a *PerformanceCharacteristic* associated with a *Function* that is **allocated to** the *Component* **described by** the current *Section*,
  - ■ **describes** a *Function* that is **allocated to** a *Component* **described by** the current *Section*,
  - ■ **describes** a *PerformanceCharacteristic* that is **specified** by a **Stand-alone** *Requirement* that **specifies** the *Component* (A *Requirement* is **Stand-alone** if it **specifies** a *PerformanceCharacteristic* that is **described by** a *Section* that **is contained by** the current *ReportDocument* and does not have the **specifies** relationship to a *Function* that **motivates** the *PerformanceCharacteristic.*), or
  - ■ for non-functional requirements Sections, **describes** a *Requirement* which **specifies** the *Component* **described by** the current *Section.*
- ❑ Column Two: The Title of the *Section* identified in Column One.
- ❑ Column Three: The Project Unique ID of:
  - ■ each *Function* associated with the *Section* referenced in Column One that has a "shall" statement in its Description and is **allocated to** the reference *Component* and of every *Requirement* printed in the *Section* referenced in Column One or printed in a numbered subsection below the *Section* referenced in Column One, and each **Stand-alone** *Requirement* that **specifies** the *Component* and the *PerformanceCharacteristic* **described by** the *Section* referenced in Column One, or
  - ■ each *Requirement* that is **described by** the *Section* listed in Column One, if that *Section* is a non-functional requirements section.

>PRINT VRS

This *ReportBlock* prints the **Standard Section Titling,** the **Standard Description** of the each element **described by** the *Section*. If that element is a *VerificationRequirement*, it also prints a three column table listing information about every *Section* that either **describes** a *Func-*

tion/*Requirement* that is **verified by** the current *VerificationRequirement* or that **describes** a *Function* which is **specified by** a *Requirement* that is **verified by** the current *VerificationRequirement* or that **describes** a *PerformanceCharactertistic* which is **motivated by** a *Function* which is **specified by** a *Requirement* that is **verified by** the current *VerificationRequirement* or that **describes** a *PerformanceCharactertistic* which is **specified by** a *Requirement* that is **verified by** the current *VerificationRequirement*.

- ❑ Column One: The Number of the *Section*.
- ❑ Column Two: The Title of the *Section*.
- ❑ Column Three: The ProjectUniqueID of each *Function*/*Requirement* in the *Section* which is **verified by** the current *VerificationRequirement*.

This *ReportBlock* also prints the Verification Level assigned to every *VerificationMethod* that is a **verification method for** the current *VerificationRequirement*.