



# MSR-Dokumentation

## Diagnose

### Strukturelle Grundlagen

**MSR-AG-Diagnose**, Herbert Klein, TZKOM

# Inhaltsverzeichnis

	Inhaltsverzeichnis	2
	Tabellenverzeichnis	4
	Präliminarien	5
<b>1</b>	<b>Einführung</b>	<b>6</b>
1.1	Vorhandene Diagnose-Dokumente	6
1.2	Mögliche Einsatzszenarien für die MSRDIA DTD bzw. ASAP2D.DTD	7
1.2.1	Lebenszyklus	7
1.2.2	Details zu den Phasen Systemlastenheft bis Kalibrierung	9
1.2.3	Zusammenwirken ASAP2D und MSRDIA	10
1.2.3.1	Entwurf der MSRDIA DTD	11
1.3	Übernahme von Daten aus anderen Instanzen bzw. Datenbanken	11
1.3.1	Generierung von Übersichten	12
1.3.2	Generierung verschiedener Dokumentansichten	12
<b>2</b>	<b>Übersicht über Diagnose-Informationen</b>	<b>14</b>
<b>3</b>	<b>Detaillierte Struktur der Diagnose-Information in MSRDIA</b>	<b>17</b>
3.1	Generelle Dokumenteninformation	17
3.2	Globale Systemidentifikation	17
3.3	Allgemeine Informationen zum Gerät/System	17
3.4	Laboraufbauten	17
3.5	Informationsspeicher	17
3.6	Fehlermanagement	18
3.7	Variantenbeschreibung	18
<b>4</b>	<b>Arbeitsgruppen-interne Informationen</b>	<b>20</b>
4.1	Noch zu klärende Detailpunkte	20
4.2	Geklärte Detailpunkte	21
4.3	Fertigstellungskriterien für MSRDIA.DTD	21
<b>5</b>	<b>Glossar</b>	<b>22</b>
<b>6</b>	<b>Elementbeschreibung</b>	<b>23</b>
6.1	mnemonic	23

	<p>Diagnose msrdia[minus ]-sp</p> <p>Kapitel:           Inhaltsverzeichnis</p>	<p>Seite:    3/30 Datum:   25.05.2000 Status:   WD</p>
---	--	--

6.2	<a href="#">data-pos</a>	23
6.3	<a href="#">data-length</a>	23
6.4	<a href="#">scaling-type</a>	23
	<a href="#">Dokumentverwaltung</a>	24
	<a href="#">Literaturverzeichnis</a>	26
	<a href="#">Index</a>	27
	<a href="#">Technische Begriffe</a>	28

	Diagnose msrdia[minus ]-sp Kapitel: Inhaltsverzeichnis	Seite: 4/30 Datum: 25.05.2000 Status: WD
---	--	--

## Tabellenverzeichnis

Tabelle 1	Lebenszyklusszenario	8
-----------	----------------------	---



## Präliminarien

Projektbeteiligte  
Firmen

### MSR-AG-Diagnose [msrdia]

Name Rollen	Abteilung	Adresse	Kontakt
Detlef Staffe, Porsche Experte			
Rudi Mayer, Bosch Experte			
Eckard Jakobi, Siemens			
Hans-Jürgen Sonnleitner, Siemens			
Uwe Bless, Daimler-Benz			
Jörg Schiller, DEBIS ASAP-Experte			
Holger Kollmeier, Daimler-Benz Experte			Holger.Kollmeier@pcd.daimler-benz.com
Oliver Marcks, VW			
Helmut Gengenbach, Bosch			
Bernhard Weichel, Bosch			
Herbert Klein, TZ-KOM DTD Architect			

Versionsinformation

Dokumentteil	Herausgeber			
	Firma	Version	Status	Anmerkungen
25.05.2000	Herbert Klein, TZKOM			
<a href="#">Details siehe Nr. 1, Seite 24</a>	msrdia	1.6	WD	

	Diagnose msrdia[minus]-sp	Seite: 6/30 Datum: 25.05.2000 Status: WD
	Kapitel: Vorhandene Diagnose-Dokumente	

# 1 Einführung

Die Arbeitsgruppe spezifiziert eine *msrdia.dtd* um die im Zusammenhang mit [Definition Diagnosejob S. 22](#) Diagnose zu handhabenden Informationen zu analysieren und zu strukturieren. Ob diese DTD dauerhaft besteht und wie ihre Beziehung zu den anderen MSR bzw ASAP DTDs gestaltet wird, ist noch offen.

## 1.1 Vorhandene Diagnose-Dokumente

Die Diagnose-Information umfaßt folgende, heute in der Regel als eigenständige Papiere verteilte Dokumente.

**System-Entwicklungsdokumentation (Diagnose)** Das ist die Dokumentation zur Entwicklung von Steuergeräten und Systemen mit Schwerpunkt Diagnose, z.B. Dokumentation der Diagnose-Eigenschaften eines Steuergerätes bzw. eines Systems.

Diese System-Entwicklungsdokumentation existiert für jedes diagnosefähige System im Fahrzeug ggf. separat<sup>1</sup>. Ein solches System kann auch eine einzelne Komponente sein<sup>2</sup>.

Zielgruppe dieser Dokumentation ist der Steuergeräte bzw. der System-Entwickler.

Dieser Bereich wird in *msrdia.dtd* abgedeckt.

**Definition Diagnose im engeren Sinne (Onboard Fehlererkennung) S. 22** Diagnose im engeren Sinne (Eigendiagnose) wird nicht in System-Entwicklungsdokumentation (Diagnose) beschrieben. Diese wird im Pflichtenheft für die SG-Software dargestellt.

**Datenversorgung für den Diagnosetester** Diese Daten werden vom Diagnosetester gelesen und verarbeitet. Es ist also die Bereitstellung der Daten für die Durchführung der Testerkommunikation.

Diese Datei stellt damit auch eine Beschreibung der Diagnosemöglichkeiten des Steuergerätes dar.

Diese Datenversorgung wird abgedeckt durch die *asap-diagnose.dtd*. Die Informationen müssen in die Entwicklungsdokumentation übernommen werden können. Andererseits sind Szenarien denkbar, bei der aus der System-Entwicklungsdokumentation Daten zu übernehmen sind (vgl. [Topic 1.2.3 Zusammenwirken ASAP2D und MSRDIA S. 10](#)).

**Produktions-Entwicklungsdokumentation (Diagnose)** Das ist die Dokumentation zur Entwicklung von Fertigungseinrichtungen in der Produktion (z.B. Bandende). Dabei sind Informationen aus der Entwicklungsdokumentation enthalten, diese werden ergänzt<sup>3</sup> um produktionsspezifische Belange (wie Vorschläge für Prüfabläufe) usw.

<sup>1</sup> Es ist zu prüfen, wie aus einzelnen Dokumenten ein Gesamtdokument für das übergeordnete System automatisch zusammengestellt werden kann.

<sup>2</sup> vgl. Komponentenstruktur in MSRSYS.DTD

<sup>3</sup> Diese Ergänzung wird verantwortlich vom Entwickler eingetragen.

	Diagnose msrdia[minus ]-sp	Seite: 7/30 Datum: 25.05.2000 Status: WD
	Kapitel: Lebenszyklus	

Diese Produktions-Entwicklungsdokumentation kann für verschiedene Systeme, Produktionsschritte, Fertigungswerke separat existieren.

Die Dokumentation kann ggf. nach Hard- Software getrennt werden.

Zielgruppe dieser Dokumentation ist der Entwickler von Produktionseinrichtungen.

Dieser Bereich wird in *msrdia.dtd* in der Weise abgedeckt, daß alle Informationen beinhaltet sind, die zwischen Produktion (Fahrzeug, und auch Steuergeräte) und Steuergeräte-entwicklung ausgetauscht bzw. verabredet werden.

**Tester-Entwicklungsdokumentation (Diagnose)** Das ist die Dokumentation zur Entwicklung von Testsystemen für die Kundendienstwerkstätten. Dabei sind Informationen aus der Entwicklungsdokumentation enthalten, diese werden aber ergänzt<sup>4</sup> um Belange der Werkstätten bzw. des Testerherstellers.

Diese Tester-Entwicklungsdokumentation existiert übergreifend für Fahrzeugbau-reihen, bzw. jedes System ggf. separat.

Die Dokumentation kann ggf. nach Hard- Software getrennt werden.

Zielgruppe dieser Dokumentation ist der Testerhersteller.

Dieser Bereich wird in *msrdia.dtd* in der Weise abgedeckt, daß alle Informatio-nen beinhaltet sind, die zwischen Testerhersteller und Steuergeräte-entwicklung ausgetauscht bzw. verabredet werden.

**Reparaturleitfaden für Kundendienstwerkstatt** Das ist die Dokumentation, die in der Service-Werkstatt verwendet wird, um [Definition Diagnose im weiteren Sinne \(Offboard Fehleranalyse\) S. 22](#) Diagnose im weitere Sinne auszuführen. Hier fließen Informationen aus der Entwicklungsdokumentation ein wie Fehlerbaum-Analyse, Schaltpläne, Systembeschreibungen, Codes, Umrechnungen usw.

Zielgruppe dieser Dokumentation ist die Kundendienstwerkstatt. Sie wird vom Fahrzeughersteller erstellt.

Dieser Bereich wird durch die *J2008.DTD* behandelt.

Dieser Bereich wird in *msrdia.dtd* aus der Sicht des System-Entwicklers abgedeckt.

Der Schwerpunkt der *msrdia.dtd* ist die Systementwicklungsdokumentation (Diagnose). Die DTD muß alles enthalten, was zwischen Entwicklung, Produktion und Kundendienstorgani-sation ausgetauscht wird. Die Pflege liegt bei der System-Entwicklung. Die *msrdia.dtd* kann auch verwendet werden, um Rückflüsse an die Entwicklung zu geben. Diese Rückflüsse sollten in separaten Instanzen erfolgen, um definierte Verantwortlichkeiten sicherzustellen.

## 1.2 Mögliche Einsatzszenarien für die MSRDIA DTD bzw. ASAP2D.DTD

### 1.2.1 Lebenszyklus

Folgendes Lebenszyklusszenario wird betrachtet:

<sup>4</sup> Diese Ergänzung wird verantwortlich vom Entwickler eingetragen.

**Tabelle 1: Lebenszyklusszenario**

Phase	Beschreibung	Diagnose-Info
Systemkonzept Fahrzeug	Es wird das Gesamtkonzept des Fahrzeuges festgelegt, inklusive der Entwicklungsprozesse	Grundsätzliche Diagnose-Themen (Protokoll, verwendete Normen, verwendete Prozesse, zu kommunizierende Informationen)
Vernetzung	Umfänge der <i>MSRNET.DTD</i>	Gegenseitige Referenzierung von Elementen: <ul style="list-style-type: none"> <li>• Vorgabe eindeutiger Identifier</li> <li>• Datalink-Layer ist ein CAN-Netzwerk</li> </ul>
<i>Systemlastenheft</i> Fahrzeug mit Komponentenstruktur	Umfänge der <i>MSRSYS.DTD</i> mit Komponenten, Signalen, Eigendiagnose, Fehlersituationen usw.	<i>Diagnose-Lastenheft</i> ergänzt als eigenständiges Dokument das <i>Systemlastenheft</i> .  Es fallen formale Diagnoseinformationen an. Bedeutung einer Diagnose-Instanz kann beginnen (Service-IDs, Protokolle, Geräteadressen, Fehlercodes)
Software-Funktionen	Funktionsumfang der SG-Software mit <b>Definition Diagnose im engeren Sinne (Onboard Fehlererkennung) S. 22</b> Diagnose im engeren Sinne (Fehleridentifikation, Testerkommunikation, Fehlermanagement)	Es fallen formale Diagnoseinformationen an. Bedeutung einer Diagnose-Instanz kann beginnen (Service-IDs, Protokolle, Geräteadressen, Fehlercodes)
Kalibrierung/Applikation	Ergänzung der der SG-Software um Fahrzeugspezifische Anpassungen.	Es entstehen z.B. Schwellen für Diagnose-Funktionen.  Es entstehen Steuergeräte Speicherinhalte, die in das Steuergerät transportiert werden. Dafür werden die Testerkommunikationseinrichtungen verwendet.
Produktionsvorbereitung		Es werden Diagnoseabläufe (Diagnose-Jobs) festgelegt.  Steuergeräte Speicherinhalte, werden in das Steuergerät transportiert. Dafür werden die Testerkommunikationseinrichtungen verwendet.

	Diagnose msrdia[minus]-sp	Seite: 9/30
	Kapitel: Details zu den Phasen Systemlastenheft bis Kalibrierung	Datum: 25.05.2000 Status: WD

**Tabelle 1 (Forts.): Lebenszyklusszenario**

Phase	Beschreibung	Diagnose-Info
Werkstatt - Kundendienst		Es werden Diagnoseabläufe (Diagnose-Jobs) festgelegt.  Steuergeräte-Speicherinhalte, die in das Steuergerät transportiert werden. Dafür werden die Testerkommunikationseinrichtungen verwendet.

## 1.2.2 Details zu den Phasen Systemlastenheft bis Kalibrierung

Im folgenden wird ein beispielhaftes Szenario skizziert (Systemneuentwicklung auch beim Systemlieferant), welches sich in den Phasen Systemlastenheft bis Kalibrierung bewegt. Das *Diagnose-Lastenheft* wird dabei mehrfach zwischen den beteiligten Stellen ausgetauscht und um Informationen ergänzt. Für andere Szenarien (wie Folgeprojekte) gelten ähnliche Aussagen.

- Der Fahrzeughersteller erstellt ein *Diagnose-Lastenheft*, in dem beschrieben ist, welche Informationen über die Diagnose-einrichtungen zu kommunizieren sind. Im Wesentlichen sind das Tabellen, von Services die zunächst nur eine grobe Beschreibung der geforderten Information enthalten. Es sind aber bereits alle Spalten für die vollständige Information angelegt, auch wenn einige davon leer bleiben. Der Fahrzeughersteller trägt die Information ein, die er weiß (z.B. nur die Spalte "Beschreibung").
- Das *Diagnose-Lastenheft* geht an den System-Zulieferer. Dort wird die Steuergeräte-software entwickelt (*System-Design*). Dabei werden entsprechend der Anforderungen aus dem *Diagnose-Lastenheft* Variablen, Kenngrößen usw. angelegt. Diese stehen dann auch als **<msrsw>** Datei zur Verfügung. Diese korrespondiert mit dem Diagnose-Lastenheft z.B. über die vorgesehene Spalte für den **<short-name>** der Steuergeräte-Variablen bzw. Parameter.
- Nach dem die Spalte<sup>5</sup> für den **<short-name>** ausgefüllt ist, können weitere Detailinformationen (z.B. Quantisierung, Maßeinheiten, Wortgrößen) aus z.B. einer *MSRSW-Datei* automatisch hinzugefügt werden.

Damit entsteht ein Dokument für den Fahrzeughersteller, das die Umsetzung des System-Zulieferers detailliert beschreibt. Dabei wird auch eine Korrespondenztabelle von Anforderung und Vorschlag generiert, welche alle vorgenommenen Änderungen bzw. Ergänzungen enthält. Das Dokument geht zurück zum Fahrzeug-Hersteller.

- Der Fahrzeug-Hersteller prüft das Ergebnis und kann Änderungswünsche eintragen. Dies geschieht z.B. dadurch, daß er den Eintrag in der Spalte für den **<short-name>** löscht oder ändert<sup>6</sup>.

<sup>5</sup> Es ist anzustreben, daß zwischen allen beteiligten Partnern die short-names vereinbart werden. Allerdings bleibt es möglich, jeweils eine eigene Spalte für den Kurzname des Herstellers bzw. des Zulieferers vorzusehen. Die Automatismen für die Dokumentpflege werden allerdings komplizierter.

	Diagnose msrdia[minus]-sp	Seite: 10/30
	Kapitel: Zusammenwirken ASAP2D und MSRDIA	Datum: 25.05.2000 Status: WD

- Das Dokument wird auch an weitere betroffene Stellen (z.B. Testerhersteller) gegeben und auch dort überprüft. Sie können als gleichberechtigter Partner das Dokument so ändern wie oben beschrieben.
- Die vorherigen Schritte können wiederholt werden, bis das System vollständig festgelegt ist.
- Das *Diagnose-Lastenheft* kann im folgenden weiter ausgewertet werden, z.B. durch
  - Erzeugung mehrerer Sprachvarianten
  - Erzeugung von Rohlingen für Testerkonfiguration oder Header-files für das Steuergerät
  - Erzeugung von Prüflisten

### 1.2.3 Zusammenwirken ASAP2D und MSRDIA

Diagnose gehört zur Entwicklungs- bzw. Systemdokumentation. Der Schwerpunkt der beiden DTDs ist.

*ASAP2D DTD* Diese Instanz ist ein Ergebnis der Entwicklung, welches die Kommunikation mit dem Tester behandelt und zur Testerkonfiguration eingesetzt wird. Die Ordnungskriterien in der DTD orientieren sich an der technischen Vollständigkeit der Testerinformationen. Sie sind daher sehr detailliert und datenbezogen.

*MSRDIA DTD* Diese Instanz ist ein sowohl entwicklungsbegleitende Dokumentation als auch ein Ergebnis der Entwicklung, welches die Diagnoseaspekte dokumentiert, Anforderungen beschreibt und sich in ihrer Struktur am menschlichen Leser orientiert. Informationen aus anderen DTDs sind zuweilen verdichtet (z.B. in Form von Übersichtstabellen).

Einsatzmöglichkeiten während der Entwicklung sind:

- Informale bzw. semiformale Beschreibung der Diagnoseaspekte, in einer am Entwickler orientierten Ordnung und Zusammenstellung in der *MSRDIA DTD*.
- Im Laufe des Prozesses können einige der semiformalen Informationen durch formale Informationen aus *MSRSSW DTD*, *MSRSYS DTD*, *MSRNET DTD*, *ASAP2D DTD* ergänzt, ersetzt bzw. detailliert werden (z.B. die grob beschriebenen Antworten von *ReadDataByLocalIdentifier* werden durch die Labels der MSRSW-Instanz ergänzt).
- Diese semiformalen Informationen können beispielsweise auch verwendet werden, um vorgefüllte Instanzen von *ASAP2D DTD* (wie z.B. SG-spezifische Anteile) zu erzeugen.

Einsatzmöglichkeiten am Ende der Entwicklung sind:

- Man könnte z.B. in **<part-type-spec>** ein Element **<diagnosis>** anbringen, welches formale Verweise auf eine ASAP-Diagnose-Instanz bzw. informelle Diagnosebeschreibungen erlaubt.
- Damit könnte man Dokumente erzeugen, die in sich geschlossenen Informationen aus beiden Welten enthält.

<sup>6</sup> Dazu muß er inzwischen vom Zulieferer die Liste der verfügbaren Informationen bekommen haben, z.B. als MSRSW-Datei.

	Diagnose msrdia[minus ]-sp	Seite: 11/30
	Kapitel: Übernahme von Daten aus anderen Instanzen bzw. Datenbanken	Datum: 25.05.2000 Status: WD

- Die gesamte Dokumentation kann in sich plausibilitätsgeprüft werden.
- Der Anwender möchte z.B. beim Lesen in einem Hypertextsystem zwischen MSR und *ASAP-Diagnose* über gemeinsame Referenzbrowser zugreifen können.

### 1.2.3.1 Entwurf der MSRDIA DTD

Alle diese Szenarien bedingen einen geordneten Datenaustausch zwischen den Instanzen. Es kann kein verbindlicher Prozeß angegeben werden, wie die Entwicklung abläuft. Daher wird ein Konzept verfolgt, welches hinsichtlich des Datenaustausches flexibel ist, und anhand der Erfahrungen aus ersten Anwendungen verfeinert bzw. korrigiert werden kann.

Der Entwurf der *MSRDIA DTD* orientiert sich daher an den vorliegenden Diagnose-Lastenheften und versucht eine formale Verbindung zu Instanzen anderer DTDs zu etablieren.

## 1.3 Übernahme von Daten aus anderen Instanzen bzw. Datenbanken

Es tritt immer wieder der Fall auf, daß die zu erfassende Information als Extrakt aus Instanzen anderer DTDs (z.B. *msrsys.dtd* bzw. *msrnet.dtd* bzw. *msrsw.dtd* *asap-diagnose.dtd*) oder anderweitig vorhandene Datenbanken zu gewinnen ist, bzw. damit in Zusammenhang steht.

Für diesen Fall wird ein neuer Mechanismus (MSR Query) vorgeschlagen, der die Einbettung bzw. die Darstellung von Zusammenhängen in DTD-unabhängiger Weise ermöglicht. Dieser ist gekennzeichnet durch:

- Die Parameter der Datenübernahme wie Semantik (Name), mögliche Argumente usw. werden angegeben.
- Die Ergebnisse der Datenübernahme stehen in einer generischen, präsentationsorientierten Struktur welche die Darstellung der Information als Tabellen, Listen usw. erlaubt. Dies bedeutet, daß bei einer Datenübernahme nicht die Originalstruktur, sondern eine präsentationsorientierte Darstellung übernommen wird.
- Durch Verwendung der generischen Struktur ist die Datenübernahme allgemein anwendbar. Die Darstellung der Daten in Tabellen, Listen usw. kann zunächst interaktiv erfolgen, in dem die Struktur (ähnlich wie bei einer Textverarbeitung) manuell angelegt und auch ausgefüllt wird. Um das Ergebnis einer automatisierten Datenübernahme reproduzierbar zu halten, ist es sinnvoll auch ein Hinweis auf die Datenquelle (Dateiname, Dateiversion, etc.) mit einfließen zu lassen.
- Gesteuert über die angegebene Semantik der Datenübernahme ist es nun möglich, in der manuell angelegten Struktur die Daten zu prüfen, zu korrigieren und auch zu erweitern. Voraussetzung hierfür ist ein spezifisches Verarbeitungsprogramm, welches die manuelle Struktur kennt (z.B. Anordnung der Spalten in einer Tabelle) und darauf arbeitet.
- Dieses Verarbeitungsprogramm kann auf Basis von generischen SGML-Werkzeugen (z.B. MetaMorphosis, Balise, OmniMark, W3C-DOM) oder auch über die Programmierschnittstelle von SGML-Editoren ausgeführt realisiert werden.
- Ein solches Verarbeitungsprogramm kann auch eingesetzt werden, um Daten zu überprüfen bzw. zu extrahieren. In frühen Phasen ist es auch möglich, die Ergebnisse

	Diagnose msrdia[minus]-sp	Seite: 12/30
	Kapitel: Generierung verschiedener Dokumentansichten	Datum: 25.05.2000 Status: WD

manuell vorwegzudefinieren, um sie bei Verfügbarkeit der Datenquelle zu ersetzen bzw. zu überprüfen (vgl. [Topic 1.2.2 Details zu den Phasen Systemlastenheft bis Kalibrierung S. 9](#)).

- Dieser Mechanismus kann auch verwendet werden, um dokumentintern Übersichten zu erstellen ([Topic 1.3.1 Generierung von Übersichten S. 12](#)).
- Wichtig ist, daß die Datenübernahme als Bestandteil der Dokumentenwartung verstanden wird, und nicht als Bestandteil der finalen Formatierung. Das bedeutet, daß nach Ausführung einer Übernahme wieder manuell in das Dokument eingegriffen werden kann. Verbindlich sind damit die Ergebnisse der Übernahme, wie sie im Dokument (**<result-p>** usw.) enthalten sind.

Das bedeutet auch, daß Dokumente nach Ausführung einer Übernahme ggf. erneut versioniert werden müssen.

- In DTDs, welche **<msr-query>** unterstützen, sollte auch **<msr-processing>** enthalten sein. Dieses dient zur Aufnahme von Verarbeitungsprotokollen usw. Damit wird nachvollziehbar, was in dem Dokument automatisch manipuliert ist.

### 1.3.1 Generierung von Übersichten

Die Informationen, die z.B. aus anderen Instanzen übernommen werden, können beim Formatieren einer *msrdia.dtd*-Instanz in tabellarischer Darstellung zusammengefaßt werden. Beispiele hierfür sind:

- Fehler mit (codes, arten, Umgebungsdaten)
- Meßwerte mit den Labels aus der Software
- Stellerlisten
- Listen mit Umrechnungsformeln

Für die Erzeugung derartiger Übersichten kann ebenfalls **<msr-query>** verwendet werden.

### 1.3.2 Generierung verschiedener Dokumentansichten

Für die Generierung verschiedener Dokumentansichten gibt es folgende Möglichkeiten:

- Ein Formatierer erzeugt die Dokument-Ansicht unter Verwendung der spezifischen Elemente in der DTD.
- Es wird für jede Ansicht manuell ein neues Dokument angelegt<sup>7</sup>, welches über **<msr-query>** die geforderten Informationen einbezieht. Der Level der Datenübernahme kann dabei als Argument an **<msr-query>** mitgegeben werden, so daß zur Laufzeit eingestellt werden kann. Dadurch wird es möglich, den Detaillierungsgrad von Dokumenten zielgruppenspezifisch anzupassen.
- Es ist möglich, in der DTD den Elementen Informationen anzuhängen, in welchen Dokumentansichten sie dargestellt werden sollen. Dies ist noch weiter zu untersuchen (todo).

<sup>7</sup> Dieses fungiert dann sozusagen als "Steuerdokument"

	Diagnose msrdia[minus ]-sp  Kapitel: Generierung verschiedener Dokumentansichten	Seite: 13/30 Datum: 25.05.2000 Status: WD
---	---	---

- Es gibt auch Fälle, bei denen eine Zusammenfassung nicht automatisch generiert werden kann, weil die Abstraktionsgrade der Informationen zu unterschiedlich sind.

	Diagnose msrdia[minus]-sp	Seite: 14/30
	Kapitel: Übersicht über Diagnose-Informationen	Datum: 25.05.2000 Status: WD

## 2 Übersicht über Diagnose-Informationen

Im Folgenden werden Informationen aufgelistet, wie sie in heute vorliegenden Dokumenten "System-Entwicklungsdokumentation (Diagnose)" gefunden werden.

Generell gilt:

- Für alle Informationen können auch Randbedingungen formuliert sein.
- Es kann Bezug genommen werden auf Variablen bzw. Kenngrößen im Steuergerät. Dieser Bezug kann dazu führen, daß Informationen (z.B. aus *MSRSW.DTD* Dateien) übernommen werden. Diese Bezüge können auch als Auswahlkriterien für die zu übernehmenden Datenlexikon-Informationen dienen.
- Es ist zu unterscheiden zwischen Anweisung und Information. Anweisung sagt, was getan werden muß. Die Information begründet die Anweisung und gibt damit Hinweise zur Realisierung. So kann z.B. die Anweisung lauten: "setze dieses Bit auf 0". Die Information kennzeichnet dieses Bit als "reserviert für Weiterentwicklung". Der SG-Entwickler würde durch die Information veranlaßt, die "0" als Festwert abzulegen und nicht im Programmcode als Literalkonstante einzuführen.
- Der Zusammenhang zwischen den Informationen der Diagnose-Services und der Implementierung im Steuergerät ist herzustellen.

Im einzelnen handelt es sich um folgende Informationen:

- Änderungsverwaltung **<admin-data>**
- Einführung, "Allgemein" **<introduction>** mit Prosatext und auch Verweisen auf andere Dokumente, z.B. Normen.
- Systemkurzbeschreibung des System (ca. 1 Seite) inklusive Diagnose-Aspekte. Diese kann als Extrakt aus *msrsys dtd* übernommen werden.
- Systemidentifikation (**<global-system-identification>**) als allgemeine Beschreibung, wie das Steuergerät oder das System zu identifizieren ist. Dies kann z.B. auch die Beschreibung von Aufklebern sein oder besonderen konstruktiven Merkmalen sein, die eine Vorabidentifizierung des Gerätes durch einen Menschen ermöglicht. Erst wenn diese Vorabidentifizierung erfolgt ist, kann ein Tester angeschlossen werden, um eine elektronische Identifizierung durchzuführen. Diese Identifikation greift nur, wenn eine einzelne Komponente behandelt werden soll.
- Grobe Beschreibung der diagnoserelevanten Fahrzeugarchitektur in **<dia-architecture>**. Hierin können Blockschaltbilder, Steckerbelegungen usw. angegeben werden. Die Information ist vergleichbar mit **<architecture>** einer *MSRSYS dtd* und kann ggf. auch aus einer solchen Instanz gewonnen werden.
- Aufgabenspezifische Labor-Aufbauten **<dia-lab-setting-spec>** als Beschreibung der Vorbedingung für die Durchführung von Diagnose-Aufgaben. Beispiele hier für sind Anschlußbeschaltung (mindestens aufzuschaltende Signale und deren Werte), Restbussimulation.
- Aufbau von Informationsspeichern (**<information-memory-spec>**) wie Fehlerspeicher usw.

	Diagnose msrdia[minus]-sp  Kapitel: Übersicht über Diagnose-Informationen	Seite: 15/30 Datum: 25.05.2000 Status: WD
---	--	---

- Unterstützte Geräte im Fahrzeug, z.B. Kontroll-Lampe, Kombi-Instrument (**<dia-devices>**).

Diese Geräte werden auch hinsichtlich der diagnoserelevanten Art der Nutzung beschrieben. Dabei kann schon grundsätzlich zwischen verschiedenen Klassen von Diagnosegeräten unterschieden werden (**<dia-device-class>**). Beispiel für solche Klassen sind "Blinklampe", "KP2000".

- Beschreibung der diagnoserelevanten Pins, z.B. pin-Nummer, Belastung der Diagnose-Leitung usw. **<communication-ports>**
- Informationen zum Protokoll (**<dia-protocol>**) (man könnte auch sagen: DataLink-Layer?). Hier werden unterschiedliche Möglichkeiten beschrieben, um Daten auszutauschen (z.B. Analogwert oder KWP2000)
  - Timing-Verhalten als Konkretisierung der in der Norm angegebenen Spielräume **<dia-protocol-timing>**
  - Kommunikationsaufbau **<dia-communication-setup>**(z.B. Adressierungsart, Übertragungsraten Headeraufbau usw.).
  - Kommunikationsabbau **<dia-communication-shutdown>**
  - Weitere Informationen zur Kommunikation:
    - management Verfahren **<dia-communication-management>**Tester present...
    - Fehlerbehandlung **<dia-communication-error-handling>**
    - generelle Struktur von Services **<dia-dia-service-layout>**
    - protokollweit vordefinierte Responses **<dia-response-codes>**
  - **<dia-protocol-services>** beschreiben Services, die zur Verwaltung der Kommunikation verwendet werden (z.B. Verändern von Kommunikationsparameter. Services die für mehrere Protokolle gelten werden in **<DIA-APP-SERVICE-SPEC>** beschrieben.
- Eine flexible Kapitelstruktur, die es erlaubt, **<chapter>** und Service-beschreibungen beliebig zu mischen, um eine angemessene Strukturierung der Dienstbeschreibung zu gewährleisten. Die Strukturierung kann über ein **<..class>** Element erfolgen. Eine solche Strukturierung ist z.B.
  - Diagnosedienste die der Mechaniker verwendet, um Fehler am Gesamtsystem [Definition Diagnose im weiteren Sinne \(Offboard Fehleranalyse\) S. 22](#)
    - Steuergeräte- und Fahrzeugidentifikationsdaten
    - Ausgabe von Fehlerspeicherinhalten  
Allgemeine Anforderung an den Fehlerspeicher (z.B. Batteriepufferung usw.)
    - Löschung von gespeicherten Informationen
    - Auslesen von Betriebsdaten
    - Auslesen von Festwerten (z.B. Schwellwerte) im Steuergerät
    - Beschreibung und Start von Download-Routinen im SG.
    - Aktivierung von Stellelementen im passiven Zustand
    - Beeinflussung von Stellelementen im aktiven Zustand

	Diagnose msrdia[minus]-sp Kapitel:           Übersicht über Diagnose-Informationen	Seite:    16/30 Datum:   25.05.2000 Status:   WD
---	--	--

- Steuergeräte-Programmierung bzw. Codierung
- Start von steuergerätespezifischen Routinen (z.B. Kalibriervorgang auslösen)
- obd-services beschreiben durch Gesetzgebung normierte Services. Diese haben eine spezielle Kommunikationsstruktur und standardisierte Inhalte.
- Diagnosespezifische Bedingungen **<dia-conditions>** beschreiben Bedingungen (z.B. Betriebszustände, Temperaturen, Drehzahlen) des Systems, die erfüllt sein müssen, daß mit dem Gerät überhaupt kommuniziert bzw. Diagnose durchgeführt werden kann bzw. der Service zur Verfügung steht.  
 Diese Bedingungen können ausführlich beschrieben oder aber unter Bezug (**<dia-condition-ref>**) auf Bedingungen anderer **<devices>** definiert werden.
- **<dia-task>** Wirkungsketten bzw. Aufgaben, die unter Verwendung von diagnose-services bzw. Eingriffen von außen (z.B. Klimaanlage einschalten) ablaufen. Hier können auch Prüfanleitungen untergebracht werden<sup>8</sup>.
- Systemimmanente Informationsspeicher in dem im laufenden Betrieb anfallende Informationen aufgesammelt werden, die z.B. über Services ausgelesen werden kann (**<information-memories>**)
- Datenlexikon-Informationen **<sw-data-dictionary>** als Zusammenfassung von Informationen aus der *msrsw.dtd*.

<sup>8</sup> diagnose-task wird zuweilen auch als diagnose-job bezeichnet.

	Diagnose msrdia[minus]-sp Kapitel: Informationsspeicher	Seite: 17/30 Datum: 25.05.2000 Status: WD
---	---	---

### 3 Detaillierte Struktur der Diagnose-Information in MSRDIA

Im Folgenden wird eine detaillierte Struktur der Diagnose-Information beschrieben, wie sie über eine *msrdia.dtd* realisiert würde. Diese DTD dient lediglich zur präsentationsorientierten Strukturierung der Information. Daher ist z.B. eine automatische Bedatung eines Testers über die *msrdia.dtd* nicht vorgesehen.

#### 3.1 Generelle Dokumenteninformation

Die generelle Dokumentinformation folgt den allgemeinen MSR-Prinzipien (vgl. *[Externes Dokument: MSR-Applikations-profil / URL: / relevante Stelle: ]* mit **<short-name>** **<project-data>**, **<admin-data>**, **<introduction>**).

#### 3.2 Globale Systemidentifikation

Systemidentifikation (**<global-system-identification>**) als allgemeine Beschreibung, wie das Steuergerät oder das System zu identifizieren ist. Dies kann z.B. auch die Beschreibung von Aufklebern sein oder besonderen konstruktiven Merkmalen sein, die eine Vorabidentifizierung des Gerätes durch einen Menschen ermöglicht. Erst wenn diese Vorabidentifizierung erfolgt ist, kann ein Tester angeschlossen werden, um eine elektronische Identifizierung durchzuführen. Diese Identifikation greift nur, wenn eine einzelne Komponente behandelt werden soll.

#### 3.3 Allgemeine Informationen zum Gerät/System

In **<general-product-data-1>** werden allgemeine Informationen zum System angegeben, um einen Überblick zu bekommen. Diese Informationen können als Extrakt aus **<TT TYPE="Product">msrsys.dtd</TT>** entnommen werden.

#### 3.4 Laboraufbauten

Hier können aufgabenspezifische Labor-Aufbauten **<dia-lab-settings>** beschrieben werden. Diese spezifizieren Vorbedingung für die Durchführung von Diagnose-Aufgaben. Beispiele hier für sind Beschaltung der des Steuergerätes, mindestens aufzuschaltende Signale und deren Werte, Restbussimulation.

#### 3.5 Informationsspeicher

In der vorliegenden DTD wird ein allgemein anwendbares Modell für übertragbare Informationsblöcke in **<information-memory-spec>** angegeben. Diese Informationen werden unabhängig vom Fehlerspeicher gesammelt und sind über Diagnosefunktionen abrufbar. Das Modell ist gekennzeichnet durch folgende Eigenschaften:

	Diagnose msrdia[minus]-sp	Seite: 18/30
	Kapitel: Variantenbeschreibung	Datum: 25.05.2000 Status: WD

- es können beliebig viele Informationsspeicher spezifiziert **<information-memory>**. Jeder dieser Speicher ist gekennzeichnet durch **<long-name>**, **<short-name>** und Klasse (**<information-memory-class>**). Information-memory-class sind u.a. Crash-Speicher, Alarm-Speicher.
- Die Position und die Größe des Informationsblockes wird mit **<information-memory-location>** und mit **<information-memory-size>** angegeben.
- Die Einträge in den Informationsblöcken (information-memory-entry) werden sequentiell definiert und durch **<information-memory-entry-class>** klassifiziert. Z.B. CAN, interne Ereignisse, Sensoren, Schalter, etc.

Die Größe des Eintrages z.B. Word, Byte, Bit etc. wird durch **<data-type>** definiert.

Das Element **<data-pos>** ist der Zugriffsindex auf den Datentyp des Entry, z.B. bei data-type = Byte kann **<data-pos>** als Bitinformation zwischen 0 und 7 liegen. Handelt es sich bei dem Informationsspeicher um einen homogenen Aufbau d.h. der **<data-type>** ist bei allen Einträgen identisch, kann **<data-pos>** wie oben beschrieben verwendet werden. Sollte ein inhomogener Aufbau existieren, so ist die **<data-pos>** als Bitinformation bezogen auf den gesamten Informationsspeicher anzugeben.

## 3.6 Fehlermanagement

Es gibt einen Service (*readDiagnosticTroubleCode*), der zum Auslesen von Fehlerinformationen (aus dem Fehlerspeicher) dient. Davon Bestandteil sind Fehlercodes und Fehlerarten. Die Pfadbeschreibung für Fehlercode und Fehlerart sollten sinnvollerweise aus den Funktionsbeschreibungen in *msrsw.dtd* extrahiert werden können. Dort muß ja beschrieben werden, wie [Definition Diagnose im engeren Sinne \(Onboard Fehlererkennung\) S. 22](#) Fehler erkannt und verwaltet werden.

Diese Fehlercodes sind in der *asap-diagnose.dtd* zwar bedatet, und mit einem Satz beschrieben. Die dahinterliegenden Fehlerbäume (also die Bedingungen, unter denen diese Fehler diagnostiziert werden) sind dort jedoch nicht enthalten. Die Bedatung der *msrsw.dtd* mit Fehlercodes (applizierbare Festwerte) sollte eigentlich aus der *asap-diagnose.dtd* heraus erfolgen.

In der Struktur **<fault-management-spec>** können mehrere Fehlerpfade (**<fault-path>**) definiert werden. Ein Fehlerpfad ist eine Überwachungseinheit für z.B. ein Sensor mit Verkabelung oder ein Steller mit Verkabelung oder eine indirekte Überwachung von Komponenten z.B. *KAT* oder *EEPROM*.

Jedem Fehlerpfad wird ein Fehlercode zugeordnet z.B. P0100.

## 3.7 Variantenbeschreibung

In der *msrdia* Beschreibung müssen auch variantenspezifische Informationen festgehalten werden können. Es wurde beschlossen folgende Struktur in der DTD zu berücksichtigen: **<general-project-data>** **<variant-spec>**.

Dies hat folgende Vorteile gegenüber einer nichtsemantischen Modellierung:

- Es kann mit xref auf einzelne Varianten referenziert werden.

	Diagnose msrdia[minus ]-sp Kapitel: Variantenbeschreibung	Seite: 19/30 Datum: 25.05.2000 Status: WD
---	---	---

- Einheitliches Layout der Variantenbeschreibung über alle msr-dia Dokumente
- Es bietet die Möglichkeit über einen separaten Prozess die Inhalte automatisch füllen zu lassen.

Innerhalb der detaillierten Diagnosebeschreibungen werden keine semantischen Strukturen verwendet. Z.B. können verschiedene Varianten durch jeweils eigene Spalten in CALS-Tabellen behandelt werden.

	Diagnose msrdia[minus]-sp	Seite: 20/30
	Kapitel: Noch zu klärende Detailpunkte	Datum: 25.05.2000 Status: WD

## 4 Arbeitsgruppen-interne Informationen

### 4.1 Noch zu klärende Detailpunkte

- Aus Gründen der Kapitelnummerierung muss überlegt werden ob die **<DIA-PROTOCOL-SERVICE-SPEC>** auf die Ebene von **<DIA-APP-SERVICE-SPEC>** zu legen.
- Wird bei der Definition von Dia-Services ein Dia-Service-ref benötigt, um doppelte definitionen zu vermeiden
- Muß MSR-Query auch bei anderen Elementen ausserhalb von NCOI möglich sein, z.B. der Generierung von Dia-Services und anderen Listen.
- Wie kann die Anwendung von **<msr-query>** (z.B. die verfügbaren Abfragen) zwischen den Partnern kommuniziert werden? (Hier geht es um einen ev. gemeinsame Entwicklung der MSR-Query Programme, nicht um den Datenaustausch zwischen den MSR Partnern)
 

11.6.1999: todo Nach einer Erfahrungsphase sollte hierzu eine weitere AG einberufen werden.
- Wie kann die Übereinstimmung von Responses mit der Implementierung im Steuergerät sichergestellt werden. Gibt es z.B. in MSRSW Mittel, um Service-Aufbauten zu beschreiben.
 

11.6.99 MSRSW bietet keine expliziten Mittel, um Services zu beschreiben. Es ist jedoch zu untersuchen, in wieweit die vorhanden **<msr-query>** genutzt werden können, um Service-Beschreibungen gegen Ergebnisse aus der SW-Entwicklung zu vergleichen.
- Einbau der Diagnose-Anteile in **<Function-description>** und Spezifikation von Fehlerpfaden und Bedingungen. Soll **<information-memory-spec>** nicht komplett in *MSRSW.DTD* übernommen werden.
 

Fehlerspeicher-Einträge als Objekte in der *MSRSW.DTD* anbringen ?

11.6.99 Die Fehlerpfade wurden vorläufig modelliert und in eine eigene Struktur gebracht. Die Übernahme nach msrsw ist noch offen.
- Sichtenkonzept
 

11.6.99 hierzu wurden Aussagen gemacht ([Topic 1.3.2 Generierung verschiedener Dokumentansichten S. 12](#)). Weitere Arbeiten notwendig.
- Normbezug mit dem Wunsch, die referenzierten Normen auch bei der Dokumentaufbereitung (Präsentation) einflechten zu können.
 

11.6.99 Die Normen müssen als SGML erfaßt werden. Dann können sie mit **<msr-query>** übernommen werden. TODO der Fall mit **<normative-name>**, welcher nicht in **<result-chapter>** usw. enthalten ist, muß noch geklärt werden. Praxiserfahrung ist notwendig.
- Anwendung der Ergebnisse auf "\$2C" - wie können temporäre Datenpakete behandelt werden.

	Diagnose msrdia[minus]-sp	Seite: 21/30
	Kapitel: Fertigstellungskriterien für MSRDIA.DTD	Datum: 25.05.2000 Status: WD

11.6.1999 Da nun die Services nicht mehr semantisch modelliert sind, muß geprüft werden, ob man mit der NCOI-Beschreibung und **<msr-query>** das Problem behandeln kann.

- **<information-memory-spec>** behandeln.

## 4.2 Geklärte Detailpunkte

- Formale Anteile in NCOI einbetten

11.6.1999 Das kann nun mit **<msr-query>** behandelt werden.

- Umgang mit Glossaren, Definitionen, Abkürzungen

11.6.1999 Behandlung über **<msr-query>**.

- Wo und wie wird das Servicelayout beschrieben.

11.6.1999 Service-layout wird unter der Kommunikationsbeschreibung der Protokolle (**<dia-com-spec>**) im Element **<dia-service-layout>** beschrieben.

- Wo werden die speziellen Anforderungen an ein Transportprotokoll beschrieben?

11.6.1999 Diese werden unter **<dia-protocol>** beschrieben. Es ist Aufgabe des Experten, die am besten passende Stelle festzulegen (ggf. **<add-spec>** verwenden).

- Es wird ein zentraler Pool von negativen Responses benötigt. Diese werden von verschiedenen Requests verwendet. Durch die Spiegelung der Identifikationsoption und der Service-Id sind die positiven Requests immer unterschiedlich. Vorschlag:n-Pool unter **<dia-service-layout>** modellieren.

Es ist noch zu klären in wie weit n-responses und p-responses sowohl bei **<dia-srv-param>** als auch bei **<dia-service>** vorhanden sein muß. Es scheint auf jeden Fall sinnvoll zu sein dies im **<dia-srv-param>** vorzusehen.

11.6.1999: es werden keine spezifischen Elemente für Responses angelegt, da eine formale Beschreibung von Services und Service-Layout nicht erfolgt. Daher können Responses in **<chapter>** beschrieben werden. Das ermöglicht auch das Referenzieren. Erhöhte Anforderungen können über msr-Query behandelt werden.

## 4.3 Fertigstellungskriterien für MSRDIA.DTD

- DTD ist anhand von zwei existierenden Lastenheften ([Mitarbeiter Detlef Staffe, Porsche S. 5](#), bzw. [Mitarbeiter Holger Kollmeier, Daimler-Benz S. 5](#)) erfolgreich verifiziert.
- DTD enthält genügt den MSR-Richtlinien
- **<msr-query>** ist verifiziert durch Beispiele, in denen die Abfrage und das erwartete Ergebnis spezifiziert ist. Beispielhafte Scripts wäre hilfreich.
- Protokolldokumente nach *[ / Norm: Keyword 2000 German Implementation / URL: / relevante Stelle: ]* sind in *msrdia.dtd* darstellbar, insbesondere die verbalen Anteile.

	Diagnose msrdia[minus]-sp	Seite: 22/30
	Kapitel: Glossar	Datum: 25.05.2000 Status: WD

## 5 Glossar

**Diagnose im engeren Sinne (Onboard Fehlererkennung)** Diagnose im engeren Sinne ist die Fähigkeit eines Steuergerätes oder eines Systems, Fehler an sich selbst, an Sensoren und Aktuatoren bis hin zum Gesamtsystem zu erkennen, auszuwerten und zu speichern. Diese werden der [Definition Kommunikationsschnittstelle \(z.B. Testerkommunikation\) S. 22](#) Kommunikationsschnittstelle (z.B. zur Testerkommunikation) zur Verfügung gestellt.

**Diagnose im weiteren Sinne (Offboard Fehleranalyse)** Diagnose im weiteren Sinne ist eine Tätigkeit, die unter Ausnutzung der über die [Definition Kommunikationsschnittstelle \(z.B. Testerkommunikation\) S. 22](#) Kommunikationsschnittstellen bereitgestellten Informationen/Aktionen (vgl. [Definition Diagnoseservice S. 22](#)) die Ursache von Fehlfunktionen von Komponenten und Systemen analysiert. Dabei können die bereitgestellten Informationen kombiniert und geeignete Schlüsse gezogen werden.

**Diagnoseservice** Ein atomarer Dienst(kleinste Informationseinheit) eines Steuergerätes, den ein *Testsystem* anstoßen kann. Er ist z.B. gekennzeichnet durch einen Identifikator, Parameter und die möglichen Antworten.

**Diagnosejob** Kombination von [Definition Diagnoseservice S. 22](#) Diagnoseservices zur Durchführung einer bestimmten Aufgabe am Gesamtsystem. Diese Kombination erfolgt im Testsystem.

**Diagnose** Dies ist ein unscharfer Überbegriff, welcher für sich alleine nicht aussagefähig ist, und daher im vorliegenden Dokument für die Gesamtheit von [Definition Diagnose im engeren Sinne \(Onboard Fehlererkennung\) S. 22](#), [Definition Diagnose im weiteren Sinne \(Offboard Fehleranalyse\) S. 22](#), [Definition Kommunikationsschnittstelle \(z.B. Testerkommunikation\) S. 22](#) verwendet.

**Kommunikationsschnittstelle (z.B. Testerkommunikation)** Ist das Protokoll mit allen Parametern die benötigt werden, um die Kommunikation zwischen einem Testsystem und einem Steuergerät betreiben zu können. Sie umfaßt *physical layer*, *datalink layer* und *application layer*.

Diese Schnittstelle bezieht sich nicht nur auf traditionelle Verfahren (K-Leitung, CAN usw.) sondern auch auf die Anwendung von Meßtechnik (z.B. Prüflampe, Spannungsmessgerät, Oszilloskop) zur Diagnose.

### Varianten

**Varietencodierung** ist die Eigenschaft eines Steuergerätes durch vorgesehene Eingriffe (z.B. Umprogrammierung einer Speicherzelle, Beschaltung eines Pins usw.) eine bestimmte Funktionsausprägung annehmen zu können. Es kann also eine der im Steuergerät enthaltenen Varianten ausgewählt werden.

In der *MSRSYS.DTD* wird **<variant-coding>** verwendet, um die Zuordnung von Varianten zu Variantenmerkmalskombinationen und deren Einfluß auf die Variantenauswahl (Sachnummernbildend oder Komponentenintern über äußere Einflüsse).

	Diagnose msrdia[minus ]-sp Kapitel:        scaling-type	Seite:    23/30 Datum:   25.05.2000 Status:   WD
---	---	--

## 6            Elementbeschreibung

### 6.1           mnemonic

#### Description

Zusätzliche Bezeichnung des Diagnoseservices, z.b. bei ReadECUIdentification ist dies "REI".

### 6.2           data-pos

#### Description

Startposition des Parameters in der Nutzinformation. Einheiten in der Nutzinformation werden ab 1 gezählt. Bei "ReadECUIdentification" Serviceparameter "Identification Option" ist dies "2". "1" ist grundsätzlich die ID des Services. Ist **<data-length>** und **<scaling-type>** nicht angegeben, so gilt wird "**<data-length>=1**" und "**<scaling-type>=Byte**".

### 6.3           data-length

#### Description

Anzahl der Informationseinheiten der Nutzinformation. Die Informationseinheit wird durch ein separates Scaling festgelegt.

### 6.4           scaling-type

Scaling Basis Type für **<data-length>** z.B. Byte, Word.

#### Description



## Dokumentverwaltung

### Änderungs- übersicht

Gesamt	Dokumentteil	Nr.	Änderung	Grund	Bezug
22.4.98		7	Ersterstellung in der AG		Inhalt
22.7.98		6	Mitschrieb der Sitzung		Inhalt
05.11.98		5	Mitschrieb der Sitzung		Inhalt
28.1.1999		4	Ergebnisse der Besprechung eingearbeitet	Erweiterung der Einsatzszenarien	Inhalt
11.6.1999		3	Ergebnisse der Besprechung eingearbeitet	nochmal Erweiterung der Einsatzszenarien, Informationsspeicher, Detailliertere Dokumentation.	Inhalt
			Beispielszenario (vom Ssystemlastenheft zur Kalibrierung) eingefügt		Inhalt
			Modell für Diagnose-Service - Semantische Modelleierung durch NCOI ersetzt	wird nun mit msr-query behandelt	Inhalt
			Diagnostic-Conditions bei Service und bei Diagnostic device angeordnet	Die Bedingung kann global oder auch service-bezogen sein.	Inhalt
			Beschreibung für MSR-Query detailliert	Das ist das wichtigste Ergebnis überhaupt.	Inhalt
			OBd-Services, Compu-Method, bit-codings usw. als eigenständige Kapitel entfernt	Die Anordnung dieser Kapitel wollten wir nicht festschreiben. Sie kann nun via msr-query überall erfolgen, wo notwendig.	Inhalt
			Offene Punkte liste <a href="#">Topic 4.1</a> <a href="#">Noch zu klärende Detailpunkte S. 20</a> aktualisiert, Arbeitsgruppeninterne Ergebnisse in ein Kapitel zusammengefasst.	Verbesserung der Übersichtlichkeit.	Inhalt
22.09.1999		2	Kapitel <a href="#">Topic 1.3 Übernahme von Daten aus anderen Instanzen bzw. Datenbanken S. 11</a> inhaltlich überarbeitet.	Detailliertere Dokumentation.	Inhalt
25.05.2000		1	Beschreibung von DIA-APP-Service-Spec	AG-DIA 25.05.2000	Inhalt



**Versions-  
übersicht**

Dokument- teil	Datum	Herausgeber			
		Firma	Version	Status	Anmerkun- gen
ab Seite 5	25.05.2000	Herbert Klein, TZKOM			
	Änderungen 1	msrdia	1.6	WD	
	22.09.1999	Uwe Bless, Daimler-Benz			
	Änderungen 2	msrdia	1.5	wd	
	11.6.1999	Bernhard Weichel, Bosch			
	Änderungen 3	msrdia	1.4	wd	
	28.1.1999	Bernhard Weichel, Bosch			
	Änderungen 4	msrdia	1.3	wd	
	05.11.98	Bernhard Weichel, Bosch			
	Änderungen 5	msrdia	1.2	wd	
	22.7.98	Bernhard Weichel, Bosch			
	Änderungen 6	msrdia	1.1	wd	
	22.4.98	Bernhard Weichel, Bosch			
	Änderungen 7	msrdia	1.0	wd	

	Diagnose msrdia[minus ]-sp  Kapitel:            Literaturverzeichnis	Seite:    26/30 Datum:    25.05.2000 Status:    WD
---	---	--

# Literaturverzeichnis

## Normen

<b>Bezeichnung:</b> <b>URL:</b> <b>relevante Stelle:</b>	[]: Keyword 2000 German Implementation	21
--	--	----

## Externe Dokumente

<b>Bezeichnung:</b> <b>URL:</b> <b>relevante Stelle:</b>	MSR-Applikations-profil  ??	17
--	-----------------------------------	----

	Diagnose msrdia[minus ]-sp  Kapitel: Index	Seite: 27/30 Datum: 25.05.2000 Status: WD
---	---	---

## Index

### B

Beispielhaftes Szenario [9](#)

### C

CALS-Tabellen [19](#)

### D

Diagnose-Dokumente [6](#)

Diagnose-Information [14](#), [17](#)

### F

Fehlermanagement [18](#)

### K

Kommunikationsschnittstelle [22](#)

### L

Lebenszyklusszenario [7](#)

### M

MSR Query [11](#)

MSR-Query [20](#)

### S

SGML-Werkzeuge [11](#)

### T

Tester [10](#), [14](#), [17](#)

### V

Variantencodierung [22](#)

	Diagnose msrdia[minus]-sp Kapitel: SGML Elemente	Seite: 28/30 Datum: 25.05.2000 Status: WD
---	--	---

## Technische Begriffe

### Produkte

#### A

ASAP-Diagnose [11](#)

asap-diagnose.dtd [6](#), [11](#), [18](#), [18](#)

ASAP2D DTD [10](#), [10](#), [10](#)

#### C

CAN [8](#)

#### J

J2008.DTD [7](#)

#### M

msrdia [18](#)

MSRDIA DTD [10](#), [10](#), [11](#)

msrdia.dtd [6](#), [6](#), [7](#), [7](#), [7](#), [7](#), [7](#), [7](#), [12](#),  
[17](#), [17](#), [21](#)

MSRNET DTD [10](#)

MSRNET.DTD [8](#), [11](#)

MSRSSW DTD [10](#)

MSRSW-Datei [9](#)

msrsw.dtd [11](#), [14](#), [16](#), [18](#), [18](#), [20](#),  
[20](#)

MSRSYS DTD [10](#), [14](#), [14](#)

MSRSYS.DTD [8](#), [11](#), [22](#)

### Programmcode

#### B

Blinklampe [15](#)

#### E

EEPROM [18](#)

#### K

KAT [18](#)

KP2000 [15](#)

#### R

ReadDataByLocalIdentifier [10](#)

readDiagnosticTroubleCode [18](#)

## SGML Elemente

### Symbole

..class [15](#)

#### A

add-spec [21](#)

admin-data [14](#), [17](#)

architecture [14](#)

#### C

chapter [15](#), [21](#)

communication-ports [15](#)

#### D

data-length [23](#), [23](#), [23](#)

data-pos [18](#), [18](#), [18](#), [18](#)

data-type [18](#), [18](#)

devices [16](#)

DIA-APP-SERVICE-SPEC [15](#), [20](#)

dia-architecture [14](#)

dia-com-spec [21](#)

dia-communication-error-handling  
[15](#)

dia-communication-management  
[15](#)

dia-communication-setup [15](#)

dia-communication-shutdown [15](#)

dia-condition-ref [16](#)

dia-conditions [16](#)

dia-device-class [15](#)

dia-devices [15](#)

dia-dia-service-layout [15](#)

dia-lab-setting-spec [14](#)

dia-lab-settings [17](#)

dia-protocol [15](#)

dia-protocol [21](#)

DIA-PROTOCOL-SERVICE-  
SPEC [20](#)

dia-protocol-services [15](#)

dia-protocol-timing [15](#)

dia-response-codes [15](#)

dia-service [21](#)

dia-service-layout [21](#), [21](#)

dia-srv-param [21](#), [21](#)

dia-task [16](#)

diagnosis [10](#)

#### F

fault-management-spec [18](#)

fault-path [18](#)

Function-description [20](#)

#### G

general-product-data-1 [17](#)

general-project-data [18](#)

global-system-identification [14](#), [17](#)



## I

information-memories [16](#)  
information-memory [18](#)  
information-memory-class [18](#)  
information-memory-entry-class [18](#)  
information-memory-location [18](#)  
information-memory-size [18](#)  
information-memory-spec [14](#), [17](#), [20](#), [21](#)  
introduction [14](#), [17](#)

## L

long-name [18](#)

## M

msr-processing [12](#)  
msr-query [12](#), [12](#), [12](#), [12](#), [20](#), [20](#), [20](#), [21](#), [21](#), [21](#), [21](#)  
msrsw [9](#)

## N

normative-name [20](#)

## P

part-type-spec [10](#)  
project-data [17](#)

## R

result-chapter [20](#)  
result-p [12](#)

## S

scaling-type [23](#), [23](#)  
short-name [9](#), [9](#), [9](#), [17](#), [18](#)  
sw-data-dictionary [16](#)

## V

variant-coding [22](#)  
variant-spec [18](#)

## Sonstige

## A

application layer [22](#)

## D

datalink layer [22](#)  
Diagnose-Lastenheft [8](#), [9](#), [9](#), [9](#), [9](#), [10](#)

## P

physical layer [22](#)

## S

System-Entwicklungsdokumentation (Diagnose) [14](#)  
Systemlastenheft [8](#), [8](#)

## Status

## S

System-Design [9](#)

## Werkzeuge

## T

Testsystem [22](#)



Diagnose msrdia[minus]-sp	Seite: 30/30
Kapitel: Konfigurationsparameter	Datum: 25.05.2000
	Status: WD

## Konfigurationsparameter

### Unternehmen (**—company**)

DC

### Sprache (**—lang**)

Deutsch

### Ausgeben des Inhalts von Querverweisen (**—xrefcontent**)

Inhalt von Xref wird ausgegeben

### Ausgabe von 'Siehe' für Querverweise

'Siehe' wird nicht ausgegeben

### Dateinamenausgabe für Grafiken (**—figname**)

Dateinamen für Grafiken werden ausgegeben

### Verarbeitung von [width] und [height] Attribut für Grafiken (**—figdimension**)

Breite und Höhe von Grafiken werden verarbeitet

### Titelseiten Graphik (**—graphic**)

Keine Titel Grafik spezifiziert

### Logo Graphik (**—head-logo**)

msrlogo.eps

### Fixtexte Datei (**—fixtext**)

C:\Programme\Medoc\Metapage\mmapps\msrrep\lib\msrrep\_ft.xml

### Ausgabe der lokalen administrativen Daten (**—admindata**)

Lokale administrative Daten werden ausgegeben

### Datei

C:\Docs\MEDOC\MSRDIA\docs\sp\msrdia-sp-de\_V210.xml

### MetaMorphosis-Version

3.2

### Formatierer-Version

2.0 (MetaPage)

### Datum

23/01/2002 10:51:45