

Bedienungsanleitung

Titel des Dokuments

Dokumentieren von Quelltexten mit IsomlDoc

im Folgenden als **Dokument** bezeichnet

Kurztitel

Bedienungsanleitung IsomlDoc



Vorgang

externe Referenz

BZ-11586

-

Das folgende Dokument ist geistiges und urheberrechtliches Eigentum der

Name und Anschrift

IFS Informationstechnik München
Trausnitzstraße 8
81671 München

im Folgenden als **IFS** bezeichnet

Die IFS behält sich ausnahmslos alle urheberrechtlichen Ansprüche vor, welche sie gemäß der geltenden Rechtslage in der Bundesrepublik Deutschland sowie aus den jeweiligen Teilen des Werkes, für die die IFS aufgrund von erfinderischer Tätigkeit natürlicherweise die Urheberschaft besitzt, ableiten kann.

Unbeeinflusst davon sind Rechte Dritter, für die die IFS, gemäß den mit diesen Dritten vereinbarten Verträgen und Überlassungen, das vorliegende Werk ggf. auch im Auftrag erstellt hat. Eine Abtretung der Urheberschaft kann daraus dennoch nicht abgeleitet werden.

Name

Sarah Erdmann

Autorin des Dokuments

Datum

15. September 2010

Erstellungsdatum

Datum

22. Januar 2013

Datum der letzten Änderung

Nummer

2.0

Version des Dokuments

Dokumentinformationen

Status der vorliegenden Version

Bezeichnung

freigegeben

Dieses Dokument unterliegt keinem systematischen Änderungsdienst.

Beteiligte und Rollen

Vorname Nachname

Rolle

Dr.-Ing. Markus A. Stulle

Auftraggeber

Carsten Smekal

Vorhabensverantwortlicher

Roland Ott

Qualitätsbeauftragter

Dr. Tim-Oliver Paaschen

Gutachter

Bettina von Staden

Lektorin

Speicherort des Dokuments

Pfad

\\ifshome.muc.smarttrust.de\IFS\Produkte\Taurus\02 Arbeitspakete\BZ-11586
(Bedienungsanleitung isomldoc)\06 Spezifikation\BZ-11586 Dokumentieren von
Quelltexten mit IsomlDoc v2.0.doc

Inhalt

Dokumentinformationen	2
1 Einleitung	4
2 Dokumentationsanweisungen für IsomlDoc	5
2.1 Allgemeine Hinweise	5
2.2 Grundlegende Dokumentationsanweisungen	6
2.3 Dokumentationsanweisungen für Funktionen	8
2.4 Weitere Dokumentationsanweisungen	9
3 Anwenden von IsomlDoc	12
3.1 Schritt 1: Werkzeug vorbereiten	12
3.2 Schritt 2: Quelltexte bereitstellen	13
3.3 Schritt 3: Konfiguration und Aufruf des Werkzeugs	18
3.4 Schritt 4: Erstellen der kompilierten Hilfe	22
Verzeichnisse und Anhänge	23
A Referenzen	23
B Übersicht aller Dokumentationsanweisungen	23

1 Einleitung

IsomlDoc ist ein Software-Dokumentationswerkzeug für *ISOM/L-Programme* [1]. *ISOM/L-Programme* beschreiben die Details der Software-Reparatur von Fahrzeugen. Mit *IsomlDoc* können aus *ISOM/L-Programmen* automatisch HTML-Dokumente erzeugt werden, die das in den Programmen enthaltene Wissen über die Fahrzeugtherapie verfügbar machen. Dazu werden spezielle *Dokumentationsanweisungen* in den Quelltext eines *ISOM/L-Programms* eingefügt, die Erläuterungen zu bestimmten Programmelementen (Namensräume, Annotationen, Funktionen) definieren. *IsomlDoc* erstellt daraus ein übersichtliches HTML-Dokument, das einen Überblick über die Elemente des *ISOM/L-Programms* und die Einzelheiten der Fahrzeugtherapie bietet.

Gegenwärtig wird mithilfe von *IsomlDoc* aus der *ISOM/L-Bibliothek für Serviceautoren (IBS)* eine kompilierte HTML-Hilfdatei generiert. Diese beschreibt alle prominenten Namensräume und Fachfunktionen der *IBS*.

Alle Namensräume und Funktionen von ISOM/L-Programmen sollten dokumentiert sein. Undokumentierte Elemente werden von IsomlDoc bei der Dokumentationserstellung über die Taurus® Ereignisaufzeichnung protokolliert (siehe Kap. 3.3.2).

2.2 Grundlegende Dokumentationsanweisungen

Üblicherweise beginnt die Dokumentation mit einer Kurzbeschreibung, die nicht länger als ein Satz sein sollte. Darauf folgt eine detaillierte Beschreibung, die mehrere Abschnitte umfassen kann:

Quellcodeauszug

```
/**
 * @brief Kurzbeschreibung des Namensraums
 *
 * @details
 * Die vollständige Beschreibung des Namensraums soll sich auf invariante
 * Teile beschränken. Weitergehende Details sind im Nachschlagewerk zu
 * dokumentieren.
 */
```

Codeauszug: Dokumentationsanweisung Kurzbeschreibung

Längere Detailbeschreibungen können in Absätze und Kapitel untergliedert werden. Einfache Absätze trennt man durch eine Leerzeile im Kommentar. Für benannte Absätze und Kapitel stehen Dokumentationsanweisungen zur Verfügung:

Quellcodeauszug

```

/*
 * @details
 * Ausführliche Beschreibung ...
 *
 * Ein neuer Absatz beginnt nach einer Leerzeile.
 *
 * @par Überschrift des benannten Absatz
 * Ein solcher Absatz wird eingerückt dargestellt
 * und endet an der nächste Leerzeile.
 *
 * @section Kapitel_1 Titel des Abschnitts
 * Die erste Zeichenkette ist der Name des Kapitels.
 * Der Name wird nicht in der erzeugten Dokumentation angezeigt!
 *
 * @subsection Unterabschnitt1 Überschrift
 * Ein Unterkapitel
 *
 * @subsubsection Unterunterabschnitt1 Überschrift
 * und ein Unterkapitel der zweiten Ebene
 */
    
```

Codeauszug: Dokumentationsanweisungen für Kurzbeschreibungen

Auf benannte Kapitel kann man verweisen:

Quellcodeauszug

```

*
 * @ref Kapitel_1
 * Als Text für diesen Verweis wird der Titel des Kapitels verwendet.
 *
 * @ref Kapitel_1 "Alternative Bezeichnung"
 * Als Text für diesen Verweis wird der Titel des Kapitels ignoriert und
 * die Bezeichnung in Anführungszeichen verwendet.
 *
    
```

Codeauszug: Auf benannte Kapitel verweisen

Will man nicht auf ein Kapitel, sondern ein anderen Teil des Textes verweisen, so kann man dort eine Zielmarke anlegen:

Quellcodeauszug

```

* @anchor Dtc_Beispiele
    
```

Codeauszug: Zielmarke

2.3 Dokumentationsanweisungen für Funktionen

Bei der Beschreibung einer Funktion ist darauf zu achten, dass der Rückgabewert und alle Parameter beschrieben werden:

Quellcodeauszug

```
/**
 * @brief Kurzbeschreibung der Funktion
 *
 * @details
 * Die vollständige Beschreibung der Funktion
 *
 * @param input Eingabeparameter
 * @return Rückgabewert
 */
Template Create(Isom.Base.String input);
```

Codeauszug: Beschreibung einer Funktion

Auch Fehlerfälle sowie Vorbedingungen, Nachbedingungen und wichtige Invarianten sollten mit den entsprechenden Dokumentationsanweisungen beschrieben werden:

Quellcodeauszug

```
*
 * @exception Das Void-Objekt wird zurückgegeben, falls ... ein Fehler
 * auftritt.
 *
 * @pre Eine Vorbedingung könnte sein, dass der Soll-Kontext bekannt
 * sein muss
 * @post Eine Nachbedingung könnte sein, dass der Plan alle erforderlichen
 * Logistikdaten enthält
 * @invariant Eine Invariante könnte sein, dass die Liste der Steuergeräte
 * nicht verändert wird
 *
```

Codeauszug: Fehlerfälle, Vor- und Nachbedingungen, Invarianten

In der Regel sollte zu jeder Funktion ein Quelltext-Beispiel angegeben werden. Dabei ist darauf zu achten, dass dieses Beispiel mit „Kopieren und Einfügen“ übernommen werden kann:

Quellcodeauszug

```

*
* @code
* // Steuergerät mit Diagnoseadresse 16 ermitteln
* Isom.Context.FutureVehicleContext
*     fvc = Isom.Context.FutureVehicleContext.GetInstance();
* Isom.Context.EcuList fvcEcus = fvc.GetEcus();
* Isom.Context.Ecu ecu = fvcEcus.FindByEcuAddress( "16" );
*
*
* if(ecu.IsVoid().Not())
* {
*     // Programmierung hinzufügen
*     Isom.Plan.TherapyPlan tp = Isom.Plan.TherapyPlan.GetInstance();
*     Isom.Plan.TherapyPlanAction tpa
*         = tp.CreateEcuAction( "Flash", ecu );
*     if(tp.Add( tpa ).Failed())
*     {
*         // Fehlerbehandlung
*     }
* }
* @endcode
*

```

Codeauszug: Quelltextbeispiel

2.4 Weitere Dokumentationsanweisungen

In den Detailbeschreibungen können weitere Dokumentationsanweisungen verwendet werden, die besondere Unterkapitel einleiten:

Quellcodeauszug

```

*
* @remarks
* Anmerkungen
*
* @note
* Hinweis
*
* @attention
* Achtung
*
* @warning
* Warnung
*
* @see
* Verweis auf andere relevante Kapitel
*

```

Codeauszug: Einleitungen für Unterkapitel

In der Regel werden auch Autor, Gutachter und das Datum der letzten Bearbeitung dokumentiert:

Quellcodeauszug

```
* @author XY (Autor)
* @author ZZ (Gutachter)
* @date letzte Änderung am 23.09.2010
```

Codeauszug: Autor, Gutachter und Datum

Außerdem kann für eine Funktion angegeben werden, seit wann sie angeboten wird, oder ob sie nicht mehr verwendet werden sollte:

Quellcodeauszug

```
*
* @since Eingeführt zu Version 3.0.0.0
*
* @deprecated Wird seit Version 5.3.0.0 nur noch aus Kompatibilitäts-
* gründen unterstützt.
*
```

Codeauszug: Weitere Details für Funktionen

Aufzählungspunkte leitet man durch ein Minuszeichen ein. Um die Punkte zu nummerieren, verwendet man zusätzlich eine Raute (-#). Unterschiedliche Einrückungstiefen erzeugen verschachtelte Aufzählungen:

Quellcodeauszug

```
*
* - Punkt 1
*   - Punkt 1.1
*     - Punkt 1.1.1
*     - Punkt 1.1.2
*   - Punkt 1.2
*   - Punkt 1.3
* - Punkt 2
*
* -# Nummerierung 1
* -# Nummerierung 2
* -# Nummerierung 3
*
```

Codeauszug: Aufzählung und Nummerierung

In Namensräumen mit zahlreichen Funktionen bietet es sich an, Gruppierungen vorzunehmen. Eine Gruppe fasst eine Reihe von Funktionen zusammen. Diese Funktionen werden in der Übersicht (am Anfang des Namensraums) zusammen dargestellt. Mit der Anweisung @name wird eine Überschrift und eine Beschreibung der Gruppe angegeben. Die beiden Anweisungen @{ und @} markieren dann Beginn und Ende der Gruppe.

Quellcodeauszug

```
/** @name Elementmethoden aller ISOM/L-Objekte
 * Technische Methoden, die von allen Fach- und Basisobjekten unterstützt
 * werden.
 * @{
 */

/**
 * Bestimmt, ob es sich um das Void-Objekt handelt.
 *
 * @return 'true', falls dies das Void-Objekt ist, 'false' sonst.
 */
objectbound Boolean IsVoid();

/**
 * Erzeugt eine identische Kopie des Objekts.
 *
 * @return Das geklonte Objekt.
 */
objectbound DtcList Clone();

/**
 * Erzeugt ein ungültiges Objekt.
 *
 * @return Ungültiges Objekt.
 */
DtcList Void();

/** @} */
```

Codeauszug: Funktionen gruppieren

3 Anwenden von IsomlDoc

Die Anwendung von IsomlDoc erfolgt in vier Schritten:

- Schritt 1: Werkzeug vorbereiten
- Schritt 2: Quelltexte bereitstellen
- Schritt 3: Konfiguration und Aufruf des Werkzeugs
- Schritt 4: Erstellen der kompilierten Hilfe

Diese vier Schritte werden im Folgenden näher erläutert.

3.1 Schritt 1: Werkzeug vorbereiten

Das Werkzeug IsomlDoc ist Bestandteil der ISOM-Autorenwerkzeuge. Um IsomlDoc verwenden zu können, müssen über die Startmaske des Taurus® Clients die ISOM-Autorenwerkzeuge eingebunden und dann der Mandant ausgewählt werden.

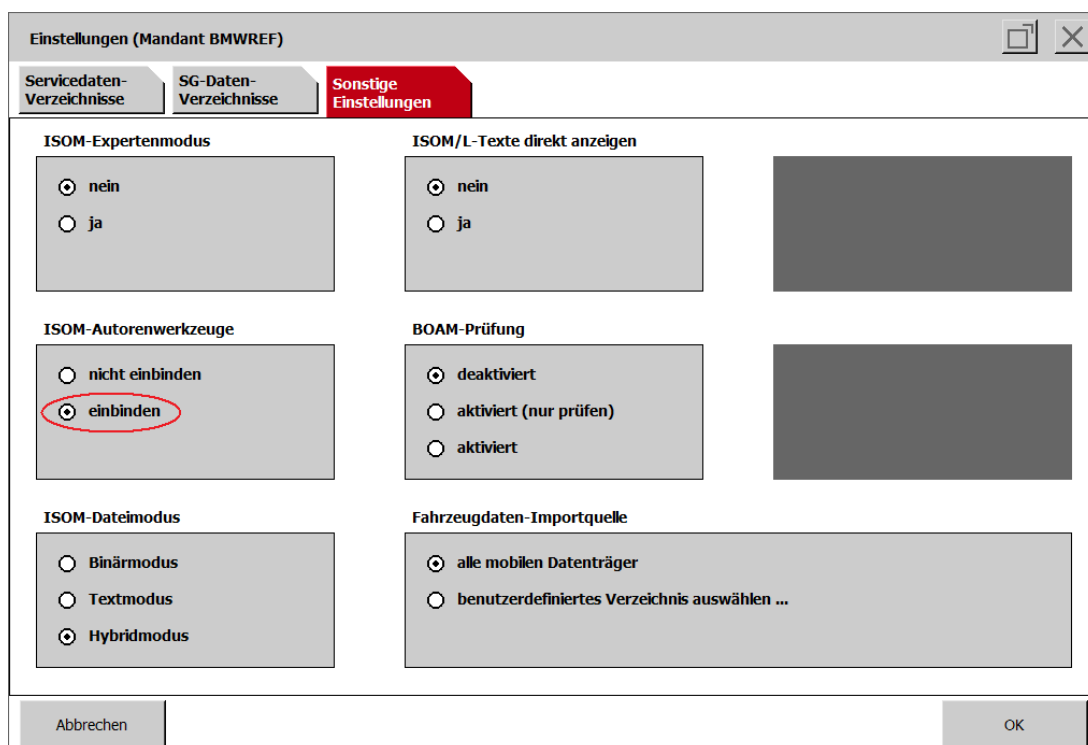


Abbildung: ISOM-Autorenwerkzeuge einbinden

3.2 Schritt 2: Quelltexte bereitstellen

Zum Erstellen der Dokumentation werden die Quelltexte der zu dokumentierenden ISOM/L-Programme benötigt. Die Quelltexte müssen die Dokumentationsanweisungen enthalten. Bevor die Dokumentation erstellt werden kann, werden noch zusätzliche Dateien benötigt:

- Eine HTML-Beschreibung des Kopfbereichs
- Eine HTML-Beschreibung des Fußbereichs
- Eine Konfigurationsdatei mit Textdefinitionen
- Optional eine Datei mit zusätzlichen Dokumentationsseiten

Diese vier Dateien werden im Folgenden kurz beschrieben.

Hinweise:

1. Alle Bilder, die in den HTML-Beschreibungen von Kopf- und Fußbereich und in den Dokumentationsanweisungen verwendet werden, müssen in dem Verzeichnis abgelegt sein, das über den Konfigurationsparameter *DocImageDir* (siehe Kap. 3.3.1) angegeben wird.
2. In den Voreinstellungen der Konfiguration werden die Bilder und die vier Dateien in dem Verzeichnis *IsomlDocSettings* unterhalb des Verzeichnisses der ISOM/L-Programme erwartet, das über den Konfigurationsparameter *OEMScriptsDir* festgelegt wird.

3.2.1 HTML-Beschreibungen von Kopf- und Fußbereich

Die beiden HTML-Beschreibungen müssen zusammengefügt ein gültiges HTML-Dokument ergeben. Die Dateinamen werden in den Konfigurationsparametern *DocHeader* und *DocFooter* angegeben. Die von *IsomlDoc* erstellte Dokumentation wird zwischen diesen beiden Bereichen eingefügt. Das folgende Beispiel zeigt den Kopf- und Fußbereich der Dokumentation der ISOM/L-Bibliothek für Serviceautoren (IBS):

Quellcodeauszug

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>${title}</title>
    <link href="${relpath$isoml.css}" rel="stylesheet" type="text/css" />
    <link href="${relpath$tabs.css}" rel="stylesheet" type="text/css" />
  </head>
  <body bgcolor="#FFFFFF">
    <table bgcolor="#FFFFFF" height="80" width="100%">
      <tr>
        <td><h1>ISOM/L-Bibliothek<br/>für Serviceautoren</h1>
          <div style="text-align: right;">
            
          </div>
        <td>
          <h2 style="text-align: center;">
            $projectnumber$
          </h2>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Codeauszug: HTML-Beschreibung des Kopfbereichs der IBS

Quellcodeauszug

```
<br/>
<table bgcolor="#F4F4F4" height="80" width="100%">
  <tr><td>Stand: $date$<br/>IFS Informationstechnik München</td></tr>
</table>
</body>
</html>
```

Codeauszug: HTML-Beschreibung des Fußbereichs der IBS

In den Beispielen werden für einige Werte Platzhalter verwendet, die von IsomlDoc beim Erstellen der Dokumentation ersetzt werden. Zulässige Platzhalter sind:

- `$date$` für das Datum der Dokumentenerstellung
- `$title$` für den Titel, der in der Textdefinition `documentTitle` angegeben ist
- `$projectname$` für die Projektbezeichnung, die in der Textdefinition `documentName` angegeben ist
- `$projectnumber$` für die Projektversion, die in der Textdefinition `documentVersion` angegeben ist

- `$relpath$` für den Pfad, in dem die Bilddateien der erstellten Dokumentation abgelegt werden. Die Ablage der Bilddateien (Ursprungsverzeichnis) für die Dokumentation kann über den Konfigurationsparameter `DocImageDir` eingestellt werden.

3.2.2 Konfigurationsdatei mit Textdefinitionen

Diese XML-Konfigurationsdatei definiert Texte, die z.B. für die Beschriftung der Navigationselemente verwendet werden. Der Dateiname wird im Konfigurationsparameter `DocTextSrc` angegeben. Das folgende Beispiel zeigt die Konfigurationsdatei der Dokumentation der ISOM/L-Bibliothek für Serviceautoren (IBS):

Quellcodeauszug

```
<?xml version="1.0" encoding="utf-8"?>
<language xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="texts.xsd"
  id="de">
  <texts>
    <text id="documentTitle">ISOM/L-Bibliothek für
    Serviceautoren</text>
    <text id="documentName">ISOM/L-Bibliothek für Serviceautoren</text>
    <text id="documentVersion">Version 6.2.0.0, IFS
    Informationstechnik GmbH</text>

    <text id="titleNamespaceIndex">Liste der Namensräume</text>
    <text id="titleFunctionIndex">Liste der Funktionen</text>
    <text id="titlePageIndex">Zusätzliche Informationen</text>
    <text id="introNamespaceTable">Hier folgen die Namensräume mit
    einer Kurzbeschreibung (wenn verfügbar):</text>
    <text id="introFunctionTable">Hier folgt die Aufzählung aller
    Funktionen mit Verweisen auf die Namensräume
    für jede Funktion:</text>
    <text id="introPageList">Hier folgt eine Liste mit
    zusammengehörigen Themengebieten:</text>
    <text id="titleDeprecatedPage">Veraltete Elemente</text>
    <text id="labelDeprecatedEntry">Element</text>
    <text id="titleActionPage">Liste der Aktionen</text>
    <text id="introActionList">Hier folgt die Übersicht aller Aktionen
    in den ISOM/L-Programmen. Zu jeder Aktion ist angegeben, in welchen
    Namensräumen Filter- bzw. Ausführungsfunktion oder
    Sichtbarkeitsterme angegeben sind.</text>
    <text id="titleEcuActionTable">Steuergerätebezogene Aktionen</text>
    <text id="titleVehicleActionTable">Fahrzeugbezogene Aktionen</text>
    <text id="actionNamespace">Namensraum</text>
    <text id="actionFilter">Filterfunktion</text>
    <text id="actionExecution">Ausführungsfunktion</text>
```

Quellcodeauszug

```

<text id="actionVisibility">Sichtbarkeit</text>

<text id="navigationMainIndex">Hauptseite</text>
<text id="navigationPageIndex">Zusätzliche Informationen</text>
<text id="Navigationnamespacemainindex">Namensräume</text>
<text id="navigationNamespaceIndex">Liste der Namensräume</text>
<text id="navigationFunctionIndex">Liste der Funktionen</text>
<text id="navigationActionIndex">Liste der Aktionen</text>

<text id="titleNamespace">Namensraum </text>
<text id="labelNamespaceLinkDetails"> Mehr... </text>
<text id="titleNamespaceDetails">Ausführliche Beschreibung</text>
<text id="titleNamespaceList">Namensräume</text>
<text id="titleFunctionList">Funktionen</text>
<text id="titleExtensionList">Erweiterungen</text>
<text id="titleFunctionDescriptions">Dokumentation der
Funktionen</text>
<text id="paragraphAuthor">Autor:</text>
<text id="paragraphDate">Datum:</text>
<text id="paragraphVersion">Version:</text>
<text id="paragraphSince">Seit:</text>
<text id="paragraphDeprecated">Veraltet:</text>
<text id="paragraphUsageHint">Verwendungshinweis:</text>
<text id="paragraphUsageExample">Verwendungsbeispiel:</text>
<text id="paragraphSee">Siehe auch:</text>
<text id="paragraphRemarks">Bemerkungen:</text>
<text id="paragraphNote">Zu beachten:</text>
<text id="paragraphAttention">Achtung:</text>
<text id="paragraphWarning">Warnung:</text>
<text id="paragraphParameter">Parameter:</text>
<text id="paragraphReturn">Rückgabe:</text>
<text id="paragraphException">Ausnahmebehandlung:</text>
<text id="paragraphPreCondition">Vorbedingung:</text>
<text id="paragraphPostCondition">Nachbedingung:</text>
<text id="paragraphInvariant">Invariant:</text>
</texts>
</language>

```

Codeauszug: Textdefinitionsdatei für IsomlDoc

3.2.3 Zusätzliche Dokumentationsseiten

In einer optionalen Datei können zusätzliche Seiten der Dokumentation angegeben werden. Der Dateiname wird im Konfigurationsparameter *DocMainPage* angegeben. Üblicherweise wird in dieser Datei die Einleitung der Dokumentation (*@mainpage*) festgelegt.

Jede Seite wird in dieser Datei in einem eigenen Blockkommentar beschrieben. Die Dokumentationsanweisungen sind die gleichen, die auch für die Dokumentation der Funktionen und Namensräume verwendet werden. Das folgende Beispiel zeigt die gekürzte Datei aus der Dokumentation der ISOM/L-Bibliothek für Serviceautoren (IBS):

Quellcodeauszug

```

/** @mainpage Serviceplattform Taurus@: ISOM/L-Bibliothek für Serviceauto-
ren
@anchor mainpage
@author IFS Informationstechnik München, <a href="http://www.ifs-
it.de/">www.ifs-it.de</a>

@section sec_intro Einführung
...

*/

/** @page pg_processmodel ISOM-Prozessmodell
Die Therapie mit ISOM folgt einem vordefinierten Zustandsmodell.

Jede Phase des Zustandsmodells lässt sich gezielt beeinflussen (inklusive
funktional leer lassen).

Die folgende Abbildung zeigt das für Autoren relevante ISOM-Prozessmodell
(Stand 07.04.2010):

@image html Prozessmodell_Autor_html.png

*/

/** @page pg_const Zeichenkettenkonstanten

@anchor const

Auflistung und Erläuterung der Zeichenkettenkonstanten für
Steuergeräteeigenschaften, die der Fachautor in den ISOM/L-Programmen
verwenden kann.
...

*/

```

Codeauszug: Zusätzliche Dokumentationsseite aus der IBS

3.3 Schritt 3: Konfiguration und Aufruf des Werkzeugs

IsomlDoc ist ein Kommandozeilenwerkzeug, das über verschiedene Konfigurationsparameter gesteuert werden kann. Alle Parameter können in der mitgelieferten Konfigurationsdatei *IsomlDoc.exe.config* eingestellt oder als Kommandozeilenparameter übergeben werden. Einstellungen auf der Kommandozeile überschreiben gleichnamige Einstellungen in der Konfigurationsdatei. Die beiden wichtigsten Parameter geben die Pfade zur ISOM/L-Bibliothek für Serviceautoren (IBS) und zu den anderen zu dokumentierenden ISOM/L-Programmen an:

Quellcodeauszug

```
IsomlDoc.exe --IsomLibDir=<Pfad zur IBS> --OEMScriptsDir=<Pfad zu anderen ISOM/L-Programmen>
```

Codeauszug: Konfigurationsparameter mit Pfaden

3.3.1 Konfiguration

Die Konfigurationsparameter von IsomlDoc sind:

Parameter	Beschreibung
Help	zeigt eine Hilfe zu den verfügbaren Kommandozeilenparametern an
DocOutputDir	Verzeichnis, in dem die erzeugte HTML-Hilfe ausgegeben wird
DocImageDir	Verzeichnis, in dem Bilder abgelegt sind, auf die in der Dokumentation verwiesen wird
DocHeader	Name der Datei mit dem Kopfbereich für jede HTML-Seite der Dokumentation
DocFooter	Name der Datei mit dem Fußbereich für jede HTML-Seite der Dokumentation
DocMainPage	Name der Datei mit der Hauptseite (Einleitung) der Dokumentation
DocTextSrc	Name der XML-Datei mit übersetzbaren Standardtexten, z.B. dem Titel der Namensraumübersicht
DocExcludeNamespaces	Semikolonseparierte Liste von Namensräumen, die nicht dokumentiert werden sollen
DocShortMemberDescriptions	gibt an, ob die Kurzbeschreibung der Funktionen in die Tabelle der Funktionen am Beginn jedes Namensraums eingefügt werden soll
IsomLibDir	Verzeichnis mit den Dateien der IBS

Parameter	Beschreibung
<code>OEMScriptsDir</code>	Verzeichnis mit den zu dokumentierenden ISOM/L-Programmen
<code>IsomModelDir</code>	Verzeichnis mit den Fahrzeugbeschreibungen
<code>Oem</code>	Name des Mandanten
<code>OEMExtensionsDir</code>	Verzeichnis mit mandantenspezifischen Erweiterungen
<code>AnnotationDataFile</code>	Datei mit den Definitionen der zulässigen Annotationen
<code>TextCategoryDataFile</code>	Datei mit den Definitionen der zulässigen Textkategorien

Tabelle: Konfigurationsparameter von `IsomlDoc`

Das folgende Beispiel zeigt eine typische Konfigurationsdatei. Die Pfade zu den Dokumentationsdateien sind alle relativ zum Parameter `OEMScriptsDir` angegeben.

Quellcodeauszug

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings" type="System.
      Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0,
      Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="IsomlDoc.Properties.Settings" type="System.
        Configuration.ClientSettingsSection, System, Version=2.0.0.0,
        Culture=neutral, PublicKeyToken=b77a5c561934e089"
        requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <IsomlDoc.Properties.Settings>
      <setting name="DocOutputDir" serializeAs="String">
        <value>?OEMScriptsDir?/doc/doc</value>
      </setting>
      <setting name="DocImageDir" serializeAs="String">
        <value>?OEMScriptsDir?/IsomlDocSettings</value>
      </setting>
      <setting name="DocHeader" serializeAs="String">
        <value>?OEMScriptsDir?/
          IsomlDocSettings/HtmlHeader.txt</value>
      </setting>
      <setting name="DocFooter" serializeAs="String">
        <value>?OEMScriptsDir?/IsomlDocSettings/
          HtmlFooter.txt</value>
      </setting>
      <setting name="DocMainPage" serializeAs="String">
```

Quellcodeauszug

```
<value>?OEMScriptsDir?/IsomlDocSettings/  
  MainPage.dox</value>  
</setting>  
<setting name="DocTextSrc" serializeAs="String">  
  <value>?OEMScriptsDir?/  
    IsomlDocSettings/Documentation_de.xml</value>  
</setting>  
<setting name="DocExcludeNamespaces" serializeAs="String">  
  <value>Example</value>  
</setting>  
<setting name="DocShortMemberDescriptions"  
  serializeAs="Boolean">  
  <value>True</value>  
</setting>  
<setting name="GlobalNamespace" serializeAs="String">  
  <value>Isom.Global.Processes</value>  
</setting>  
<setting name="Oem" serializeAs="String">  
  <value />  
</setting>  
<setting name="IsomLibDir" serializeAs="String">  
  <value>?IsomBase?\isoml</value>  
</setting>  
<setting name="IsomModelDir" serializeAs="String">  
  <value>?VehicleModel?</value>  
</setting>  
<setting name="OEMScriptsDir" serializeAs="String">  
  <value>?OemIsomL?</value>  
</setting>  
<setting name="OEMExtensionsDir" serializeAs="String">  
  <value>?OemIsomL?\..\bin</value>  
</setting>  
<setting name="AnnotationDataFile" serializeAs="String">  
  <value>?OEMScriptsDir?\AnnotationData.xml</value>  
</setting>  
<setting name="TextCategoryDataFile" serializeAs="String">  
  <value>?OEMScriptsDir?\TextCategoryData.xml</value>  
</setting>  
<setting name="SpecializationDataFile" serializeAs="String">  
  <value>?OEMScriptsDir?\Specialization.xml</value>  
</setting>  
</IsomlDoc.Properties.Settings>  
</applicationSettings>  
</configuration>
```

Codeauszug: Einstellungen d. mitgelieferten Konfigurationsdatei

3.3.2 Ausgabe

IsomlDoc verwendet die Taurus® Ereignisaufzeichnung, um relevante Ereignisse zu protokollieren. Zusätzlich werden Informationen auf der Konsole ausgegeben. Falls keine Fehler auftreten, sieht die Ausgabe wie folgt aus:

Ausgabe

```
Dokumentationserstellung beginnen...

  ISOM/L-Programme laden...

  Zusätzliche Dateien laden...

  Dokumentation erstellen...

  Dokumentation ausgeben...

Die Dokumentationserstellung ist abgeschlossen.

Die erstellten Dateien befinden sich im Verzeichnis: ...
```

Konsolenausgabe: Keine Fehler aufgetreten

Fehlermeldungen und Warnungen werden nicht nur über die Ereignisaufzeichnung protokolliert, sondern zusätzlich auf der Konsole ausgegeben. Das kann wie folgt aussehen:

Ausgabe

```
Dokumentationserstellung beginnen...

  ISOM/L-Programme laden...

  Zusätzliche Dateien laden...

  Dokumentation erstellen...

  Für 846 Funktionen und Namensräume ist keine Dokumentation
  vorhanden.

  Dokumentation ausgeben...

Die Dokumentationserstellung ist abgeschlossen.

Die erstellten Dateien befinden sich im Verzeichnis: ...
```

Konsolenausgabe: Fehlermeldung

3.4 Schritt 4: Erstellen der kompilierten Hilfe

Nachdem das Werkzeug IsomlDoc erfolgreich auf der Kommandozeile aufgerufen wurde, liegt die Dokumentation der ISOM/L-Programme in Form zahlreicher HTML-Seiten vor, die in dem Verzeichnis abgelegt wurden, das über den Konfigurationsparameter *DocOutputDir* festgelegt wird. IsomlDoc hat zusätzlich zu den HTML-Seiten auch Konfigurationsdateien für das Werkzeug *HTML Help Workshop* der Firma Microsoft [2] erstellt. Mit diesem Werkzeug können die HTML-Seiten in ein einzelnes Archiv im Format Compiled HTML Help (*.chm) überführt werden. Das Werkzeug muss dazu zunächst installiert werden. Nach der Installation wird es auf der Kommandozeile aufgerufen:

Quellcodeauszug

```
<Pfad zum HTMLHelpWorkshop>hhc.exe <DocOutputDir>/index.hhp
```

Codeauszug: Erzeugen der Hilfe mit HTML Help Workshop

Verzeichnisse und Anhänge

A Referenzen

Referenz auf

- [1] Hoffmann, Simon (IFS): BZ-9758 Handbuch *Einführung in die Programmiersprache ISOM/L*, Version 1.0, 05.02.2010.
- [2] Microsoft Download Center, HTML Help Workshop and Documentation: <http://www.microsoft.com/downloads/details.aspx?FamilyID=00535334-c8a6-452f-9aa0-d597d16580cc>, Stand: 10.09.2010 16:45 Uhr.

B Übersicht aller Dokumentationsanweisungen

Mit Ausnahme der Aufzählungsanweisungen - und -# werden alle Dokumentationsanweisungen durch das Zeichen @ eingeleitet.

Anweisung	Beschreibung	Verwendung
-	Aufzählungspunkt	muss am Anfang einer Zeile stehen. Unterschiedlich tiefe Einrückungen werden als Verschachtelung interpretiert.
-#	nummerierter Aufzählungspunkt	muss am Anfang einer Zeile stehen. Unterschiedlich tiefe Einrückungen werden als Verschachtelung interpretiert.
@a Wort	Das folgende <i>Wort</i> bezeichnet ein <i>Argument</i> und wird durch einen anderen Zeichensatz hervorgehoben.	wird im laufenden Text verwendet, um auf ein Funktionsargument zu verweisen
@anchor Marke	Die folgende <i>Marke</i> wird nicht dargestellt, kann aber von anderen Stellen der Dokumentation aus referenziert werden.	steht üblicherweise auf einer eigenen Zeile, z.B. vor einem Bild, auf das in einem Absatz verwiesen wird
@attention	markiert einen Absatz mit einem Hinweis	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
@author	markiert einen Absatz, in dem ein Verfasser angegeben wird	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.

Anweisung	Beschreibung	Verwendung
<code>@b Wort</code>	Das folgende <i>Wort</i> wird durch Fettschrift hervorgehoben.	wird im laufenden Text verwendet, um ein Wort hervorzuheben
<code>@brief</code>	markiert einen Absatz mit einer Kurzbeschreibung einer Funktion oder eines Namensraums	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@c Wort</code>	Das folgende <i>Wort</i> bezeichnet ein <i>Code-Element</i> und wird durch einen anderen Zeichensatz hervorgehoben.	wird im laufenden Text verwendet, um eine ISOM/L-Anweisung hervorzuheben
<code>@code @endcode</code>	Die Zeilen zwischen den beiden Anweisungen werden als Quelltextbeispiel betrachtet. Formatierungen (Umbrüche und Einrückungen) bleiben erhalten.	Diese Anweisungen stehen immer einzeln in eigenen Zeilen.
<code>@date</code>	Markiert eine Datumsangabe; üblicherweise die letzte Bearbeitung	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@deprecated</code>	markiert einen Absatz, in dem beschrieben wird, seit wann die dokumentierte Funktion veraltet ist.	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@details</code>	markiert einen Absatz mit der ausführlichen Beschreibung einer Funktion oder eines Namensraums	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@e (auch @em)</code>	Das folgende <i>Wort</i> wird durch kursive Schrift hervorgehoben.	wird im laufenden Text verwendet, um ein Wort hervorzuheben
<code>@exception</code>	Beschreibt eine Bedingung, unter der die Funktion ein Void-Objekt zurückgibt	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@if @endif</code>	Die Zeilen zwischen den beiden Anweisungen werden ignoriert. Die Bedingung wird momentan nicht ausgewertet.	Diese Anweisungen stehen immer einzeln in eigenen Zeilen.

Anweisung	Beschreibung	Verwendung
<code>@image <i>Format</i> <i>Dateiname</i></code>	verweist auf die Bilddatei <i>Dateiname</i> , die an dieser Stelle einzufügen ist, falls das <i>Format</i> dem Ausgabeformat (z.B. <i>HTML</i>) entspricht.	steht immer in einer eigenen Zeile. Der Text nach den ersten beiden Parametern wird als Bildunterschrift verwendet.
<code>@invariant</code>	markiert einen Absatz, in dem Invarianzen der Funktion beschrieben werden	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@li</code>	Aufzählungspunkt	muss am Anfang einer Zeile stehen. Unterschiedlich tiefe Einrückungen werden als Verschachtelung interpretiert
<code>@mainpage</code>	markiert einen Abschnitt, der den Inhalt der Hauptseite enthält. Darf nur einmal in einer Dokumentation vorkommen.	steht immer in einer eigenen Zeile. Der Text auf dieser Zeile wird als Überschrift interpretiert.
<code>@name</code>	enthält die Beschreibung für den folgenden Block. Blöcke umfassen üblicherweise mehrere Funktionen, die dann in der Liste der Funktionen (am Anfang eines Namensraums) in einem eigenen Absatz beschrieben werden. Blöcke können nicht geschachtelt werden!	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@{</code>	Anfang eines Blocks.	Diese Anweisungen stehen immer einzeln in einer eigenen Zeile.
<code>@}</code>	Ende eines Blocks.	Diese Anweisungen stehen immer einzeln in einer eigenen Zeile.
<code>@note</code>	markiert einen Absatz mit einer Anmerkung	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.

Anweisung	Beschreibung	Verwendung
<code>@p</code>	Das folgende <i>Wort</i> bezeichnet einen <i>Parameter</i> und wird durch einen anderen Zeichensatz hervorgehoben.	wird im laufenden Text verwendet, um auf einen Funktionsparameter zu verweisen
<code>@page Seitenname</code>	markiert einen Abschnitt, der den Inhalt der Seite <i>Seitenname</i> enthält	steht immer in einer eigenen Zeile. Text nach dem Parameter <i>Seitenname</i> wird als Überschrift interpretiert. Der Seitenname kann z.B. als Ziel für Verweise dienen.
<code>@par</code>	markiert einen neuen Absatz	steht am Anfang einer Zeile. Der Absatz-Inhalt beginnt immer in der folgenden Zeile. In der gleichen Zeile kann eine optionale Überschrift für den Absatz angegeben werden.
<code>@param Parametername</code>	beschreibt den Funktionsparameter <i>Parametername</i>	steht am Anfang einer Zeile. Das erste folgende Wort ist der Name des Parameters, der Rest des Absatzes ist die Beschreibung.
<code>@pre</code>	beschreibt eine Vorbedingung einer Funktion	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@post</code>	beschreibt eine Nachbedingung einer Funktion	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@ref Ziel</code>	erzeugt einen Verweis auf das <i>Ziel</i> . Das <i>Ziel</i> kann eine Marke (siehe <i>@anchor</i>) sein, ein benannter Absatz, ein Kapitel oder der Name eines dokumentierten Elements (Funktion oder Namensraum)	kann im laufenden Text verwendet werden. Folgt auf den Parameter <i>Ziel</i> Text in Anführungszeichen, so wird dieser Text als Beschriftung des Verweises verwendet; sonst wird als Beschriftung der Name des Ziels genutzt (z.B. bei Verweisen auf Funktionen) oder dessen Überschrift (z.B. bei Verweisen auf Seiten und Kapitel).
<code>@remarks</code>	markiert einen Absatz mit einer Bemerkung	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.

Anweisung	Beschreibung	Verwendung
<code>@return</code> (auch <code>@returns</code>)	beschreibt den Rückgabewert einer Funktion	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@see</code> (auch <code>@sa</code>)	markiert einen Absatz mit Verweisen („Siehe auch“)	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@section Name</code>	beschreibt das Kapitel <i>Name</i>	steht immer in einer eigenen Zeile. Der Text nach dem Parameter <i>Name</i> wird als Überschrift interpretiert. Der Kapitelname kann z.B. als Ziel für Verweise dienen.
<code>@subsection Name</code>	beschreibt das Unterkapitel <i>Name</i>	steht immer in einer eigenen Zeile. Der Text nach dem Parameter <i>Name</i> wird als Überschrift interpretiert. Der Kapitelname kann z.B. als Ziel für Verweise dienen.
<code>@subsubsection Name</code>	beschreibt das Unterkapitel <i>Name</i> der zweiten Ebene	steht immer in einer eigenen Zeile. Der Text nach dem Parameter <i>Name</i> wird als Überschrift interpretiert. Der Kapitelname kann z.B. als Ziel für Verweise dienen.
<code>@since</code>	markiert einen Absatz, in dem beschrieben wird, seit wann die dokumentierte Funktion verfügbar ist	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@subpage Seitenname</code>	markiert eine Abschnitt, der den Inhalt der untergeordneten Seite <i>Seitenname</i> enthält	steht immer in einer eigenen Zeile. Der Text nach dem Parameter <i>Seitenname</i> wird als Überschrift interpretiert. Der <i>Seitenname</i> kann z.B. als Ziel für Verweise dienen.
<code>@usagehints</code>	markiert einen Absatz mit Verwendungshinweisen	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
<code>@usageexample</code>	markiert einen Absatz mit Verwendungsbeispielen	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.

Anweisung	Beschreibung	Verwendung
@version	markiert einen Absatz, in dem eine Version angegeben wird	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.
@verbatim @endverbatim	Die Zeilen zwischen den beiden Anweisungen sollen <i>wörtlich</i> übernommen werden. Formatierungen (Umbrüche und Einrückungen) bleiben erhalten.	Diese Anweisungen stehen immer einzeln in eigenen Zeilen.
@warning	markiert einen Absatz mit einer Warnung	steht am Anfang einer Zeile. Der Absatz-Inhalt kann auf der gleichen oder der folgenden Zeile beginnen.

Tabelle: Alphabetische Liste der Dokumentationsanweisungen