

Script Engine Dokumentation

Version 1.4

2022-10-19

Inhaltsverzeichnis

1 Allgemein	4
1.1 Skriptdatei	4
1.2 Elemente	4
1.2.1 <task> Element	4
1.2.2 <define> Element	6
1.2.3 Beispiel	7
1.3 Variablen	7
1.3.1 Quelle: Skript-Variablen	8
1.3.2 Quelle: zusätzliche ini-Datei	8
1.3.3 Beispiel	8
2 Aktionen	10
2.1 Condition	10
2.2 Configure	10
2.3 Connect	11
2.4 Control	11
2.5 Create	11
2.6 Destroy	12
2.7 DestroyAll	12
2.8 Disconnect	13
2.9 Execute	13
2.10 IPC	14
2.11 Load	14
2.12 Manipulate	15
2.13 Playlist	15
2.14 Refresh	16
2.15 RefreshAll	16
2.16 Render	16
2.17 Save	16
2.18 Send	17
2.19 Start	17
2.20 StartAll	18
2.21 Stop	18
2.22 StopAll	18
2.23 Transfer	18
2.24 Wait	19
3 Parameterblock	21
3.1 action="Configure" state="BaseMethod"	21
3.2 action="Configure" state="DeviceSel"	23
3.3 action="Configure" state="ConvertConfig"	26
3.4 action="Configure" state="ExportConfig"	31
3.5 action="Configure" state="Validate"	35
3.6 action="Configure" state="StartOption"	37
3.7 action="Configure" state="all"	37
3.7.1 type="Renderer" subtype="dx9"	37
3.8 action="Create" typ="Behaviour"	37
3.9 action="Create" typ="Behaviour" subtyp="SingleClientCalib"	38
3.10 action="Create" typ="CntSpace" subtyp="3DModel"	39

3.11	action="Create" typ="file" subtype="bmp"	40
3.12	action="Execute" typ="Timer" subtype="Sleep"	40
3.13	action="Execute" typ="App"	41
3.14	action="Execute" typ="OS"	41
3.15	action="Load", "Save" typ="file" subtype="sps"	42
3.16	action="Load", "Save" typ="file" subtype="vc"	44
3.17	action="Load", "Save" typ="file" subtype="playlist"	44
3.18	action="Save" typ="file" subtype="calibuserdata"	45
3.19	action="Load" typ="file" subtype="calibuserdata"	45
3.20	action="Transfer" typ="file"	46
3.21	action="Transfer" typ="Export"	46
3.22	action="Playlist"	48
3.23	action="Render"	49
3.24	action="Manipulate" typ="Calibration" subtype="Blending"	50
3.25	action="Manipulate" typ="Calibration" subtype="CalibChange"	51
3.26	action="Manipulate" typ="Calibration" subtype="Mask"	53
3.27	action="Manipulate" typ="Calibration" subtype="Blacklevel"	54
3.28	action="Manipulate" typ="ProjectManager"	55
3.29	action="Manipulate" typ="NativeWB"	56
3.30	action="Manipulate" typ="GUI"	58
3.31	action="Send" typ="TCP" subtype="Command"	60
3.32	action="Send" typ="TCP" subtype="File"	61
3.33	action="IPC" typ="Common"	62
3.34	action="Condition"	63
3.35	action="Wait"	67
4	zusätzliche Konfigurationsdateien	68
4.1	Selektor-Parameter Liste	68

1 Allgemein

1.1 Skriptdatei

Eine Skriptdatei ist eine ANSI Textdatei im XML Format mit der Wurzelsektion **<VIOSO>**.

```
<?xml version="1.0"?>
<VIOSO>
  <File version="1.0.0" />
</VIOSO>
```

1.2 Elemente

Es gibt zwei Arten von Elementen:

1.2.1 <task> Element

Eine **<task>** Sektion definiert eine bestimmte Aktion.

Jede **<task>** Sektion muß ein **action** Attribut haben, welches die auszuführende Aktion definiert.

In Abhängigkeit von der angegebenen Aktion sind i.d.R. noch weitere Attribute zwingend anzugeben (siehe Abschnitt 2).

action	Beschreibung	Attribute	
		benötigt	optional
Condition	Dient zur Spezifikation einer bedingten Abarbeitungsaktion (siehe Abschnitt 2.1)	name	use
Configure	Dient zum konfigurieren einer bekannten Entität. (siehe Abschnitt 2.2)	name, state	use
Connect	Öffnet eine Verbindung zum angegebenen Rechner (Client). (siehe Abschnitt 2.3)	core	
Control	Zeigt an, daß ein bestimmtes Verhalten auf einem Client überwacht (kontrolliert) werden soll. (siehe Abschnitt 2.4)	typ, subtyp, core	
Create	Erzeugt eine bestimmte Entität. (siehe Abschnitt 2.5)	typ, subtyp	core, use, name
Destroy	Beendet/ zerstört eine bestimmte Entität (siehe Abschnitt 2.6)	name oder typ, subtyp	
DestroyAll	Beendet/ zerstört alle existierende Entitäten eines bestimmten Typs (siehe Abschnitt 2.7)	typ, subtyp	
Disconnect	Schließt die Verbindung zum	core	

	angebenen Rechner (Client). (siehe Abschnitt 2.8)		
Execute	Führt den spezifizierten Befehl aus. (siehe Abschnitt 2.9)	typ, subtype	use, name
IPC	Ermöglicht die Konfiguration der IPC Schnittstelel (siehe Abschnitt 2.10)	typ	use
Load	Läd eine bestimmte Datei/ Konfiguration/ etc. (siehe Abschnitt 2.11)	typ, subtype, name	use
Manipulate	Aktion zum verändern/ manipulieren von Daten wie Kalibrierungen etc. (siehe Abschnitt 2.12)	typ, subtype	use
Playlist	Konfiguriert/ manipuliert die Playlist einer Presenter Entität. (siehe Abschnitt 2.13)	name	use
Refresh	Aktualisiert eine Ausgabe/ etc. (siehe Abschnitt 2.14)	name oder typ, subtype	
RefreshAll	Aktualisiert die Ausgaben einer Entitätsklasse. (siehe Abschnitt 2.15)	typ, subtype	
Render	Rendert den aktuellen Zustand einer Renderer Entität, (siehe Abschnitt 2.16)	name	use
Save	Speichert Kalibrierungen/ Konfigurationen/ etc. (siehe Abschnitt 2.17)	typ, subtype	use
Send	Aktion zum senden von Informationen übers Netzwerk. (siehe Abschnitt 2.18)	typ, subtype	use
Start	Startet ein Verhalten/ etc. (siehe Abschnitt 2.19)	name oder typ, subtype	
StartAll	Startet alle existierende Verhalten eines bestimmten Typs. (siehe Abschnitt 2.20)	typ, subtype	
Stop	Stoppt die Verarbeitung eines Verhaltens/ etc. (siehe Abschnitt 2.21)	name oder typ, subtype	
StopAll	Stoppt die Verarbeitung aller Verhalten eines bestimmten Typs. (siehe Abschnitt 2.22)	typ, subtype	
Transfer	Aktion zum Transfer von Informationen/ Dateien. (siehe Abschnitt 2.23)	typ	use
Wait	Allgemeine Aktion zum warten auf bestimmte Ereignisse. (siehe Abschnitt 2.24)	name, state	core

1.2.2 <define> Element

Eine <define> Sektion dient zur Zusammenfassung von Parameter-Sektionen, die für bestimmte Aktionen benötigt werden. Parameter-Sektionen können auch direkt in der <task> Sektion angegeben werden. Eine <define> Sektion dient der besseren Lesbarkeit eines Scripts und der Wiederverwendung der gleichen Parameter in mehreren <task> Sektionen.

Eine <define> Sektion benötigt zwingend folgende Attribute:

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name des Parameterblocks
typ	3DModel	dient der Basisinitialisierung eines 3DModelTreat Verhaltens (siehe Abschnitt 3.1)
	App	Allgemeine Parameter zur Ausführung von Anwendungen. (siehe Abschnitt 3.13)
	BehaviourCreate	dient zum erzeugen/ starten eines bestimmten Verhaltens (siehe Abschnitt 3.8ff)
	BlendParam	dient zur Konfiguration/ Manipulation von edge blending Parametern (siehe Abschnitt 3.24)
	CalibChange	Dient zur allgemeinen Manipulation von Kalibrierungen (siehe Abschnitt 3.25)
	CalibCommerce	dient zur Konfiguration eines Export/ Convert Verhaltens (siehe Abschnitt 3.4)
	CntSpace	Parameter zur Erzeugung eines custom content space Eintrages (siehe Abschnitt 3.10)
	Common	allgemeine Parameter, verwendet für mehrere Aktionen (siehe Abschnitt 3.12ff.)
	Condition	Parameter zur Spezifikation einer bedingten Abarbeitungsaktion (siehe Abschnitt 3.34)
	CreateImage	Parameter zur Erzeugung einer Standard-Bilddatei (siehe Abschnitt 3.11)
	OS	Parameter zum ausführen von Betriebssystem spezifischen Aufgaben. (siehe Abschnitt 3.14)
	Playlist	Parameter zur Konfiguration einer Playlist (siehe Abschnitt 3.22)
	RenderPipe	dient zur Konfiguration eines Presenter Verhaltens (siehe Abschnitt 3.5)
	RenderTarget	Parameter zur Definition, wohin das Ergebnis eines Renderschrittes abgelegt werden soll. (siehe Abschnitt 3.23)

	SC_DevSel	dient zur Konfiguration des Geräteauswahl-Schrittes eines SingleClientCalib Verhaltens (siehe Abschnitt 3.2)
	Transfer	Parameter zur Durchführung einer Transfer-Aktion . (siehe Abschnitt 3.20)

1.2.3 Beispiel

Folgende einfache Scriptdatei kopiert die Datei "Calib.sps" aus dem Datenordner, nach "C:\Backup\Calib.sps".

```
<?xml version="1.0"?>
<VIOSO>
  <File version="1.0.0" />
  <task action="transfer" typ="file" use="Backup" />

  <define name="Backup" typ="Transfer" >
    <from file="Calib.sps" />
    <to file="C:\Backup\Calib.sps" />
  </define>
</VIOSO>
```

Folgende Scriptdatei tut genau das gleiche, nur das die nötigen Parameter für die Transfer Aktion direkt in der **<task>** Sektion angegeben werden.

```
<?xml version="1.0"?>
<VIOSO>
  <File version="1.0.0" />
  <task action="transfer" typ="file" >
    <from file="Calib.sps" />
    <to file="C:\Backup\Calib.sps" />
  </task>
</VIOSO>
```

1.3 Variablen

An einigen Stellen können Attribut-Werte durch Variablen definiert werden. Dies ermöglicht die Einbindung von zusätzlichen ini-Dateien bzw. die Verwendung von Skript-Parametern. In einer Skript-Datei haben Variablen grundsätzlich folgende Notation:

\$(Quelle)\$(Name)\$

- **[Quelle]**: Bezeichner der Variablen-Quelle
- **[Name]**: bezeichnet den Namen der Variable, die verwendet werden soll

1.3.1 Quelle: Skript-Variablen

Skript-Variablen sind Parameter die direkt mit dem Namen der auszuführenden Skript-Datei spezifiziert werden. Skript-Variablen haben nur für das auszuführende Skript Gültigkeit und werden als Parameter-String an die Skript-Engine übergeben.

Der Parameter-String hat folgenden Aufbau:

[Name]=[Wert]{,[Typ]};[Name]=[Wert]{,[Typ]}; ...

- **[Name]**: Name der Skript-Variable
- **[Wert]**: Wert der Skript-Variable
- **[Typ]**: optionale Angabe des Typs des Variablen-Werts
wird kein **[Typ]** angegeben, wird **[Wert]** als Zeichenkette interpretiert
mögliche Typenbezeichner:
 - **i** => Wert bezeichnet eine ganze Zahl (Integer)
 - **f** => Wert bezeichnet eine Fließkomma-Zahl (float)
 - **d** => Wert bezeichnet eine Fließkomma-Zahl mit doppelter Präzession (double)
 - **s** => Wert bezeichnet eine Zeichenkette

Im Skript können Skript-Variablen durch den Quellnamen: **ENV** abgerufen werden (siehe unten).

1.3.2 Quelle: zusätzliche ini-Datei

Diese Variablen-Quelle kann zur Einbindung von projektspezifischen Konfigurationsdateien und allgemein zur Einbindung von Selektor-Parameter Listen verwendet werden (siehe Abschnitt 4.1). Der Inhalt von zusätzlichen ini-Dateien ist persistent verfügbar. Einmal geladen, können alle nachfolgenden Skripte auf den Inhalt zugreifen.

1.3.3 Beispiel

Dieses Beispielskript lädt die ini-Datei „List.ini“, interpretiert als Selektor-Parameter Liste, und veröffentlicht den Inhalt der ini-Datei unter dem Namen `Calibs`.

```
<?xml version="1.0"?>
<VIOSO>
  <task action="load" name="Calibs" type="file" subtype="paramlist"
        use="spsL" />

  <define name="spsL" type="Common" >
    <param file="List.ini" />
  </define>
</VIOSO>
```


Die Datei „List.ini“ hat beispielsweise folgenden Inhalt:

```
<?xml version="1.0"?>
<VIOSO>
  <list>
    <element selector="el1" value="c1.sps" />
    <element selector="el2" value="c2.sps" />
  </list>
</VIOSO>
```

Wird jetzt folgendes Skript mit dem Parameter-String: use=el1; aufgerufen:

```
<?xml version="1.0"?>
<VIOSO>
  <task action="load" name="sps" type="file" subtype="sps" use="sps.Load" />

  <define name="sps.Load" type="Common" >
    <param sel="$ENV$use$" list="Calibs" />
  </define>
</VIOSO>
```

wird die Datei „c1.sps“ geladen.

2 Aktionen

2.1 Condition

Spezifiziert eine bedingte Abarbeitungsaktion (siehe Abschnitt 3.34).

Benötigte Attribute: **name**

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name der Bedingung

Optionale Attribute: **use**

Attribut	Wert	Beschreibung
use	[Bezeichner]	Name eines Parameterblocks

2.2 Configure

Dient zum konfigurieren einer bekannten Entität.

Benötigte Attribute: **name, state**

Attribut	Wert	Beschreibung	verfügbar für Entität	
			typ	subtyp
name	[Bezeichner]	Name der Entität die konfiguriert werden soll	[alle]	[alle]
state	BaseMethod	Basis Konfiguration (siehe Abschnitt 3.1)	Behaviour	3DModelTreat
	Validate Presentation	Presenter-Konfiguration (siehe Abschnitt 3.5)	Behaviour	Presenter
	DeviceSel	Geräteauswahl-Konfiguration (siehe Abschnitt 3.2)	Behaviour	SingleClientCalib
	ExportConfig	Export-Task Konfiguration (siehe Abschnitt 3.4)	Behaviour	Export
	ConvertConfig	Konversion-Task Konfiguration (siehe Abschnitt 3.3)	Behaviour	Convert
	all	allgemeiner Zustand (siehe Abschnitt 3.7)	Renderer	dx9
	StartOption	Konfiguration von Start Optionen (siehe Abschnitt 3.6)	Behaviour	Presenter

Optionale Attribute: **use**

Attribut	Wert	Beschreibung
use	[Bezeichner]	Name eines Parameterblocks

2.3 Connect

Öffnet eine Verbindung zum angegebenen Rechner (Client).

Benötigte Attribute: **core**

Attribut	Wert	Beschreibung
core	[IP]	IP-Adresse des gewünschten PCs

2.4 Control

Zeigt an, daß ein bestimmtes Verhalten auf einem Client überwacht (kontrolliert) werden soll.

Bisher werden nur nach dem Starten der Überwachen erstellte Entitäten überwacht.

Benötigte Attribute: **typ, subtype, core**

Attribut	Wert	Beschreibung
typ	behaviour	nur dieser Typ ist hier erlaubt
subtype	[Bezeichner]	alle verfügbaren Behaviour Untertypen (siehe Abschnitt 2.5)
core	[IP]	IP-Adresse des gewünschten PCs

2.5 Create

Erzeugt eine bestimmte Entität.

Benötigte Attribute: **typ, subtype**

Attribut		Beschreibung
typ	subtype	
Behaviour	Presenter	erzeugt eine Presenter-Linie (siehe Abschnitt 3.8)
	3DModelTreat	erzeugt einen 3D-Modell Treatment Scheduler (siehe Abschnitt 3.8)
	SingleClientCalib	erzeugt einen Single-Client Kalibrator Scheduler (siehe Abschnitt 3.9)
	Export	erzeugt einen Export Scheduler (siehe Abschnitt 3.8)

	Convert	Erzeugt einen Conversion Scheduler (siehe Abschnitt 3.8)
	MultiClientCalib	erzeugt einen Multi-Client Kalibrator Scheduler (siehe Abschnitt 3.8)
CntSpace	3DModel	erzeugt ein 3D Model custom content space (siehe Abschnitt 3.10)
file	bmp	erzeugt eine einfarbige Bilddatei (Bitmap) (siehe Abschnitt 3.11)
renderer	dx9	erzeugt einen DirectX 9 Renderer (siehe Abschnitt)

Optionale Attribute: **core**, **use**, **name**

Attribut	Wert	Beschreibung	verfügbar für Typ
core	[IP]	IP-Adresse des gewünschten PCs	Behaviour
use	[Bezeichner]	Name eines Parameterblocks	Behaviour , CntSpace
name	[Bezeichner]	Name der Entität (nötig wenn weitere Script Tasks auf der Entität angewendet werden sollen)	Behaviour , CntSpace

2.6 Destroy

Beendet/ zerstört eine bestimmte Entität.

Benötigte Attribute: **name** oder **typ**, **subtyp**

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name der Entität (bisher nur Verhalten)

oder

Attribut		Beschreibung
typ	subtyp	
Behaviour	[siehe Abschnitt 2.5]	beendet/ zerstört alle Entitäten eines bestimmten Typs

2.7 DestroyAll

Beendet/ zerstört alle Entitäten eines bestimmten Typs.

Benötigte Attribute: **typ**, **subtyp**

Siehe Abschnitt.: 2.6

2.8 Disconnect

Beendet die Verbindung zum angegebenen Rechner (Client).

Benötigte Attribute: **core**

Attribut	Wert	Beschreibung
core	[IP]	IP-Adresse des gewünschten PCs

2.9 Execute

Führt den spezifizierten Befehl aus.

Benötigte Attribute: **typ, subtyp**

Attribut		Beschreibung
typ	subtyp	
Timer	sleep	pausiert die Scriptverarbeitung für eine bestimmte Zeit (siehe Abschnitt 3.12)
Progress	Next	geht in der Abarbeitung eines sequentiellen Verhaltens einen Schritt vorwärts
	Back	geht in der Abarbeitung eines sequentiellen Verhaltens einen Schritt zurück
	Abort	bricht die Abarbeitung eines Verhaltens ab
App	Start	Startet eine Anwendung die sich im <i>Execute-Ordner</i> der ausführenden Anwendung befinden muß. (siehe Abschnitt 3.13)
	Restart	Ermöglicht den Neustart der ausführenden Anwendung (siehe Abschnitt 3.13)
	Shutdown	Ermöglicht das Beenden einer aktuell ausgeführten Anwendung. (siehe Abschnitt 3.13)
OS	Restart	Ermöglicht das neustarten des Betriebssystems. (siehe Abschnitt 3.14)
	Shutdown	Ermöglicht das herunterfahren des Betriebssystems. (siehe Abschnitt 3.14)

Optionale Attribute: **use, name**

Attribut	Wert	Beschreibung	verfügbar für Typ
----------	------	--------------	-------------------

use	[Bezeichner]	Name eines Parameterblocks	Timer, App, OS
name	[Bezeichner]	Name des Verhaltens	Progress, Abort

2.10 IPC

Konfiguriert/ bestückt die IPC Schnittstelle.

Benötigte Attribute: **typ**

Attribut typ	Beschreibung
Common	Allgemeine Aktionen zur Konfiguration der IPC Schnittstelle (siehe Abschnitt 3.33)

Optionale Attribute: **use**

Attribut	Wert	Beschreibung	verfügbar für Typ
use	[Bezeichner]	Name eines Parameterblocks	Common

2.11 Load

Läd eine bestimmte Datei/ Konfiguration/ etc.

Benötigte Attribute: **typ, subtype, name**

typ	Attribut subtype	Beschreibung
File	arva.ini	Läd eine spezielle Konfigurationsdatei für das ARVA Projekt
	sps	Läd eine Kalibrierungsdatei im sps Format (siehe Abschnitt 3.15)
	vc	Läd eine virtuell canvas Beschreibungsdatei (siehe Abschnitt 3.16)
	paramlist	Läd eine spezielle Parameter Liste Datei (siehe Abschnitt 3.15)
	playlist	Läd eine Playlist (im XML Format) (siehe Abschnitt 3.17)
	calibuserdata	Läd eine spezielle binär Datei mit kalibrierungsspezifischen Nutzerdaten (siehe Abschnitt 3.19)

Optionale Attribute: **use**

Attribut	Wert	Beschreibung	verfügbar für Typ
use	[Bezeichner]	Name eines Parameterblocks	File

2.12 Manipulate

Aktion zum verändern/ manipulieren von Daten wie Kalibrierungen etc.

Benötigte Attribute: **typ** (, **subtyp**)

Attribut		Beschreibung
typ	subtyp	
Calibration	Blending	Erlaubt die Manipulation der edge blending Parameter (derzeit nur für Compound-Displays verfügbar) (siehe Abschnitt 3.24)
	CalibChange	Erlaubt allgemeine Manipulationen einer Kalibrierung (derzeit nur für Compound-Displays verfügbar) (siehe Abschnitt 3.25)
	Mask	Erlaubt die Manipulation der Masken Parameter einer Einzel-Display-Kalibrierung (siehe Abschnitt 3.26)
	BlackLevel	Erlaubt die Manipulation der Blacklevel Parameter (derzeit nur für Compound-Displays verfügbar) (siehe Abschnitt 3.27)
ProjectManager		Erlaubt die Manipulation des Projekt-Managers (siehe Abschnitt 3.28)
NativeWB		Erlaubt die Manipulation von native warp fähigen Display-Treibern (Nvidia) (siehe Abschnitt 3.29)
GUI		Erlaubt die Manipulation des User Interfaces (siehe Abschnitt 3.30)

Optionale Attribute: **use**

Attribut	Wert	Beschreibung	verfügbar für Typ
use	[Bezeichner]	Name eines Parameterblocks	Calibration, NativeWB, GUI, ProjectManager

2.13 Playlist

Konfiguriert/ manipuliert die Playlist einer Presenter Entität.

Benötigte Attribute: **name**

Attribut	Wert	Beschreibung	verfügbar für Typ
name	[Bezeichner]	Name der Entität mit Playlist	Behaviour

Optionale Attribute: **use**

Attribut	Wert	Beschreibung	verfügbar für Typ
use	[Bezeichner]	Name eines Parameterblocks (siehe Abschnitt 3.22)	Behaviour

2.14 Refresh

Aktualisiert eine Ausgabe/ etc.

Benötigte Attribute: **name** oder **typ**, **subtype**

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name der Entität (bisher nur Verhalten)

oder

Attribut		Beschreibung
typ	subtype	
Behaviour	Presenter	aktualisiert die Ausgabe aller aktiver Presenter-Linien (bei Dateiausgabe werden die angezeigten Dateien neu geladen)

2.15 RefreshAll

Aktualisiert die Ausgaben einer Entitätsklasse.

Benötigte Attribute: **typ**, **subtype**

Siehe Abschnitt.: 2.14

2.16 Render

Rendert den aktuellen Zustand eines Renderers.

Benötigte Attribute: **name**

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name des Renderers

2.17 Save

Speichert Kalibrierungen/ Konfigurationen/ etc.

Benötigte Attribute: **typ**, **subtype**

Attribut		Beschreibung
typ	subtype	

File	sps	speichert eine Kalibrierungsdatei im sps Format (siehe Abschnitt 3.15)
	vc	speichert eine virtuell canvas Beschreibungsdatei (siehe Abschnitt 3.16)
	playlist	speichert die Playlist einer Presenterlinie (im XML Format) (siehe Abschnitt 3.17)
	calibuserdata	speichert eine spezielle binär Datei mit kalibrierungsspezifischen Nutzerdaten (siehe Abschnitt 3.18)

Optionale Attribute: **use**

Attribut	Wert	Beschreibung	verfügbar für Typ
use	[Bezeichner]	Name eines Parameterblocks	File

2.18 Send

Aktion zum senden von Informationen übers Netzwerk.

Benötigte Attribute: **typ, subtype**

Attribut		Beschreibung
typ	subtype	
TCP	Command	Sendet einen (einfachen) Kommandostring zum spezifizierten Client (siehe Abschnitt 3.31)
	File	Sendet eine Datei zum spezifizierten Client (siehe Abschnitt 3.32)

Optionale Attribute: **use**

Attribut	Wert	Beschreibung	verfügbar für Typ
use	[Bezeichner]	Name eines Parameterblocks	Send

2.19 Start

Startet ein Verhalten/ etc.

Benötigte Attribute: **name** oder **typ, subtype**

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name der Entität (bisher nur Verhalten)

oder

Attribut		Beschreibung
typ	subtype	

Behaviour	Presenter	startet die Ausgabe aller existierender Presenter-Linien
	3DModelTreat	startet alle existierende 3D-Modell Treatments
	Export	startet alle existierende Export Vorgänge

2.20 StartAll

Startet alle existierenden Verhalten eines bestimmten Typs.

Benötigte Attribute: **typ**, **subtyp**

Siehe Abschnitt.: 2.19

2.21 Stop

Stoppt die Verarbeitung eines Verhaltens/ etc.

Benötigte Attribute: **name** oder **typ**, **subtyp**

Attribut	Wert	Beschreibung
name	[Bezeichner]	Name der Entität (bisher nur Verhalten)

oder

Attribut		Beschreibung
typ	subtyp	
Behaviour	Presenter	stoppt die Ausgabe aller existierender Presenter-Linien

2.22 StopAll

Stoppt die Verarbeitung aller Verhalten eines bestimmten Typs.

Benötigte Attribute: **typ**, **subtyp**

Siehe Abschnitt.: 2.21

2.23 Transfer

Aktion zum Transfer von Informationen/ Dateien.

Benötigte Attribute: **typ**

Attribut	Wert	Beschreibung
----------	------	--------------

typ	file	Datei Transfer Operation (siehe Abschnitt 3.20)
	export	Transfer Operation einer vorherigen Export Task (siehe Abschnitt 3.21)

Optionale Attribute: **use**

Attribut	Wert	Beschreibung
use	[Bezeichner]	Name eines Parameterblocks

2.24 Wait

Allgemeine Aktion zum warten auf bestimmte Ereignisse.

Benötigte Attribute: **name**, **state**

Attribut	Wert	Beschreibung	verfügbar für Verhalten
name	[Bezeichner]	Name der Entität (bisher nur Verhalten)	
state	controlled	wartet bis ein bestimmtes Remote-Verhalten unter der Kontrolle des Masters ist	[alle]
	BaseMethod	wartet bis das spezifizierte Verhalten im Zustand zur grundlegender Konfiguration ist	3DModelTreat
	StaticModel	wartet bis das spezifizierte Verhalten zur Eingabe statischer Modell-Parameter bereit ist	3DModelTreat
	StdInteraction	wartet bis das spezifizierte Verhalten seinen Standard-Interaktions-Kanal anbietet	3DModelTreat, Presenter
	Validate	wartet bis das spezifizierte Verhalten in einem validiertem Zustand ist	Presenter
	Presentation	wartet bis das spezifizierte Verhalten im Presentation-Zustand ist	Presenter
	Interact.DeviceSel	wartet bis das spezifizierte Verhalten den Interaktionskanal zur Eingabe einer Geräteauswahl anbietet	SingleClientCalib
	Interact.Export	wartet bis das spezifizierte Verhalten den Interaktionskanal zur Eingabe von Export-Parametern anbietet	Export
	Interact.Convert	wartet bis das spezifizierte Verhalten den Interaktionskanal zur Eingabe von	Convert

		Konversion-Parametern anbietet	
	Finished	wartet bis das spezifizierte Verhalten beendet wurde (siehe Abschnitt 3.35)	SingleClientCalib, Export

Optionale Attribute: **core**, **use**

Attribut	Wert	Beschreibung	verfügbar für state
core	[IP]	IP-Adresse des gewünschten PCs	controlled
use	[Bezeichner]	Name eines Parameterblocks	Finished

3 Parameterblock

3.1 *action="Configure" state="BaseMethod"*

Typ des separaten Parameterblocks: **3DModel**

Verfügbare Sektionen:

- bis auf **file** und **tMethod** sind alle Attribute in **<param>** Sektion optional

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname]	Name der sps-Kalibrierungsdatei, die die zu nutzende Multi-Client Kalibrierung enthält
	tMethod	static model	es soll ein statisches 3D-Modell verwendet werden
		dynamic marker detection	bisher nicht implementiert
	model	[Bezeichner]	Name des zu verwendenden 3D-Modells (custom content space Definition)
	externAssigns	[Variable]	Referenz auf eine ini-Datei Variable auf eine Liste von Parametern zur Konfiguration einer automatischen Observer Konvertierung verwendeter Displays
	bUseModelCntSpace	[0, 1]	Verwendung der Texture-Koordinaten des verwendeten 3D-Modells als content-space
	bUseDisplMap	[0, 1]	Verwendung einer Verschiebungs-Lookup (displacement map) zur Bestimmung der tatsächlichen Display Pixel
	bGenP2WMap	[0, 1]	Erzeugung einer Display Pixel zu Welt (3D) Lookup
	bFillSmallMeshHoles	[0, 1]	kleine Löcher im 3D-Modell (Übergänge zwischen triangulierten NURBS-Flächen) sollen bei der blending Berechnung gefixt werden
	bSplitLargeMeshes	[0, 1]	Dreieckslisten sehr große 3D-Modelle sollen in kleinere Dreieckslisten zerlegt werden

	bPostProcessBlending	[0, 1]	bildbasierte Nachbearbeitung der erzeugten blend-Masken zur Verbesserung der Randdarstellung
	bNoBlending	[0, 1]	es soll keine blend-Maske berechnet werden
	bAutoObserverConversion	[0, 1]	es sollte eine Oberserver Konvertierung verwendeter Displays durchgeführt werden
	bUseCurrPrjMask	[0, 1]	bei der Erzeugung der Display Maps/blendings soll eine gesetzte Display-Maske beachtet werden
	bFillSmall3DMeshHoles	[0, 1]	kleine Löcher im 3D-Modell (Übergänge zwischen triangulierten NURBS-Flächen) sollen direkt im 3D-Modell gefixt werden
	qSmallHoleContourMax	[1 .. n]	maximale Contour-Größe eines 3D-Modell Lochs zur Detektion kleiner Löcher
	qSmallHoleSearchSz	[1 .. n]	bestimmt die Koppelweite des Suchfilters zur Detektion kleiner Löcher im 3D-Modell
	smallHoleAngle	[1 .. 360]	definiert einen Winkel in Grad zur Bestimmung der Form kleiner Löcher

```

<define name="3D.BaseParam" typ="3DModel" >
  <param file="SC_MT_1.sps"          tMethod="static model"
        model="externModel"          externAssigns="$ARVA$Assigns$"
        bUseModelCntSpace="1"         bUseDisplMap="1"
        bGenP2WMap="1"               bFillSmallMeshHoles="1"
        bSplitLargeMeshes="1"         bPostProcessBlending="1"
        bAutoObserverConversion="1"   bUseCurrPrjMask="0"
        bFillSmall3DMeshHoles="1"     bNoBlending="0"
        qSmallHoleContourMax="80"     smallHoleAngle="10"
        qSmallHoleSearchSz="1"
  />
</define>

```

3.2 action="Configure" state="DeviceSel"

Typ des separaten Parameterblocks: SC_DevSel

Verfügbare Sektionen:

- es können mehrere <display> Sektionen angegeben werden
- bis auf **tCalib** sind alle Attribute in <param> Sektion optional

Sektion	Attribut	Wert	Beschreibung
<param>	tCalib	SingleManual	manuelle Einzel-Display Kalibrierung
		OldAny	alte beliebige Oberflächen Kalibrierung
		any	beliebige Oberflächen Kalibrierung
		flat	flache Oberflächen Kalibrierung
		curved	gebogene Oberflächen Kalibrierung
		externP2C	externe geometrische Display Kalibrierung
		RoundCave	spezielle Rundleinwand Kalibrierung
		Manual	manuelle Multi-Display Kalibrierung
		Preceeding	Blending Berechnung auf Basis vorhergehender Kalibrierungen
		extern3DModel	3D Modell basierte Kalibrierung auf Basis externer Informationen
	tArrangement	strip	Automatische Display-Strip Anordnung Erkennung
		grid	Beliebige Display Anordnung
		hStrip	horizontale Display-Strip Anordnung
		vStrip	vertikale Display-Strip Anordnung
	tCamSpaceConvert	none	Keine Kameraraum Konversion
		SrcToDst	Quell zu Ziel Kameraraum Konversion (Master-Slave Kamerasystem)
	calibName	[Bezeichner]	Name (Beschreibung) der Kalibrierung
	fileName	[Dateiname]	Name der Datei wohin die Kalibrierung gespeichert werden soll (nicht immer verfügbar)
	contentRatio	free	freies Abbildungsverhältnis
		[x:y]	resultierendes

			Inhaltsabbildungsverhältnis (Bsp: 16:9)
	geometricScanSize	[float]	Faktor zur Einstellung der geometrischen Abtastweite (Bsp: 1.2)
	blendingLinearizeSz	[int]	Linearisierungsweite benötigt für die Blending Berechnung (sollte immer 0 sein)
	bHQBlending	[0, 1]	high quality (höher aufgelöste) Blendingberechnung
	bContourBlending	[0, 1]	konturbasierte Blendingberechnung
	bAdditionalLNS	[0, 1]	zusätzliche beliebige Oberflächen Kalibrierung
	bCntBasedManualMP	[0, 1]	Inhaltsbasierte manuelle Kalibrierung
	bOutsideCamView	[0, 1]	Kalibrierung auch über das Kamerabild hinaus
	bFillAbsentBubbles	[0, 1]	nicht erkannte Meßpunkte sollten extrapoliert werden
	bFixLinePattern	[0, 1]	Das Linienmuster Meßbild wird aufbereitet
	bExtLoD0	[0, 1]	Verwendung einer nicht planaren LoD0 Hypothese
	bRegulariseLNS	[0, 1]	Ausreißerelementierung
	bCalcDisplayPose	[0, 1]	Bestimmung der Display-Pose aller kalibrierten Displays
	bP2WAvailable	[0, 1]	nicht dokumentiert
	bAutoCntSpaceConvert	[0, 1]	Automatische Inhaltsraum Konversion
	bSavePartialCalib	[0, 1]	individuelles speichern einer Kalibrierung nach jedem Display
	bSetDisplayMask	[0, 1]	Der Nutzer erhält die Möglichkeit zum setzen einer Display-Maske
	bAWBCamera	[0, 1]	Eine Kamera mit aktiviertem auto white balancing wird verwendet
	bVignetteFilter	[0, 1]	Verwendung eines Filters zur Kompensation eines Vignetteneffekts
	bAutoInvisibleBlendingCrop	[0, 1]	Verwendung einer automatischen Beschneidung nicht sichtbarer Display Bereiche
	qVisiblePrjRegionExtend	[0 .. n]	Erweiterung des erkannten sichtbaren Displaybereichs in Display Pixel

	bUseMonoCamera	[0, 1]	Verwendung einer monochromen Kamera
	bUseHWBlend	[0, 1]	Generiert Skalierungsmasken um Projektionen mit Hardware Blend zu verbessern.
	bForceNewScan	[0, 1]	Wenn aktiviert, wird immer ein neuer geometrischer Scan durchgeführt.
<display>	tDevice	sd	einzelnes Display
	name	[Bezeichner]	Name des Displays
	iDeviceGroup	[Index]	Wenn kein name angegeben, bestimmt den Index (beginnend mit 1) des zu verwendenden Displays.
<camera>	tDevice	camera	einzelne Kamera
	name	[Bezeichner]	Name der Kamera
	iDeviceGroup	[Index]	Wenn kein name angegeben, bestimmt den Index (beginnend mit 1) der zu verwendenden Kamera.

```

<define name="C1.DevSel" typ="SC_DevSel" >
  <camera tDevice="camera" name="Logitech QuickCam Pro 9000" />
  <display tDevice="sd" name="D2 SyncMaster (SAM04EA)" />
  <param tCalib="flat" tArrangement="grid"
    tCamSpaceConvert="none"
    calibName="Test"
    contentRatio="16:9" geometricScanSize="2.0"
    bHQBlending="0" bContourBlending="1"
    bAdditionalLNS="0"
    bOutsideCamView="1" bFillAbsentBubbles="1"
    bFixLinePattern="1"
    bExtLoD0="0" bRegulariseLNS="0"
    bCalcDisplayPose="0" bAutoCntSpaceConvert="0"
    bSavePartialCalib="0" bSetDisplayMask="0"
    bAWBCamera="0" bVignetteFilter="0"
    bAutoInvisibleBlendingCrop="0"
    qVisiblePrjRegionExtend="30"
    bUseMonoCamera="0" bUseHWBlend="0"
    bForceNewScan="0"
  />
</define>

```

3.3 **action="Configure" state="ConvertConfig"**

Typ des separaten Parameterblocks: **CalibCommerce**

Verfügbare Sektionen:

- es können mehrere <display> Sektionen angegeben werden
- bis auf **tConvert** sind alle Attribute in <param> Sektion optional
viele Parameter aus Abschnitt 3.4 können auch hier verwendet werden

Sektion	Attribut	Wert	Beschreibung
<param>	tConvert	custom content space conversion	Konvertiert die ausgewählten Kalibrierungen in den angegebenen Content Space.
		observer conversion	Konvertiert das Content-Mapping der ausgewählten Kalibrierungen in den angegebenen Observer
		exclude compound display	Entfernt die ausgewählten Display-Kalibrierungen aus dem verwendenden Display-Compound
		add VC to P2C	Fügt die aktuelle VC der ausgewählten Kalibrierungen zum Content Mapping hinzu und setzt eine Standard-VC
		camera content space translation	Erzeugt eine Übersetzungsmatrix für die verwendete Kameras
		pseudo 3D plane	Erzeugt ein 3D Map für die ausgewählten Kalibrierungen, basierend auf planarem 3D Modell
		pseudo 3D sphere	Erzeugt ein 3D Map für die ausgewählten Kalibrierungen, basierend auf Halbkugel 3D Modell
		pseudo 3D equirectangular	Erzeugt ein 3D Map für die ausgewählten Kalibrierungen, basierend auf Halbkugel 3D Modell mit equirectangularer Verteilung
		pseudo 3D cylinder	Erzeugt ein 3D Map für die ausgewählten Kalibrierungen, basierend auf Cylinder 3D Modell
		merge calibration	Fügt ausgewählte Compound/ Supercompound Kalibrierungen zu einer Compound/ Supercompound Kalibrierung zusammen

		auto frustum	Berechnet ein mögliches Sicht-Frustum für die Displays aller ausgewählten Kalibrierungen
		clone calibration	Erzeugt ein Duplikat von den ausgewählten Kalibrierungen
		remove P2C outlier	Versucht Ausreißer im Content-Mapping der ausgewählten Kalibrierungen zu erkennen und zu entfernen.
		smooth P2C	Zeichnet das Content-Mapping der ausgewählten Kalibrierungen weich.
		PrjSpace overlap	Ermöglicht das Mapping jedes verwendeten Displays in den Display-Raum jedes Displays
		Crop Display Geometry	Ermöglicht das beschneiden der Display-Geometrie mit Hilfe einer Masken-Dateien (Bild)
		Crop Display Geometry with Projector Mask	Ermöglicht das beschneiden der Display-Geometrie mit Hilfe der aktuell gesetzten Display-Maske
		delete calibration	Ermöglicht das löschen ausgewählter Kalibrierungen
<display>	tDevice	sd	einzelnes Display
		dc	Display Compound
		sc	Super Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="pseudo 3D sphere" bNoDefaultParam="1" />
  <param0 X="-180.0" Y="180.0" Z="0.0" W="90.0" />
  <!-- X => min u    Y => max u    Z => min v    W => max v | all in degree -->
  <param1 X="1000.0" Y="0.0" Z="0.0" W="0.0" />
  <!-- X => radius    Y => radius2    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="pseudo 3D plane" bNoDefaultParam="1" />
  <param0 X="1600.0" Y="0.0" Z="900.0" W="0.0" />
  <!-- X => width    Y => ...    Z => height    W => ... -->
  <param1 X="1000.0" Y="0.0" Z="0.0" W="0.0" />
  <!-- X => distance    Y => unused    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="pseudo 3D equirectangular" bNoDefaultParam="1" />
  <param0 X="-180.0" Y="180.0" Z="-60.0" W="60.0" />
  <!-- X => min u    Y => max u    Z => min v    W => max v | all in degree -->
  <param1 X="1000.0" Y="0.0" Z="0.0" W="0.0" />
  <!-- X => radius    Y => radius2    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="pseudo 3D cylinder" bNoDefaultParam="1" />
  <param1 X="1000.0" Y="0.0" Z="0.0" W="0.0" />
  <!-- X => radius    Y => radius2    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="add vc to P2C"
        bNoDefaultParam="0"    bPostProcess="1"
  />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="custom content space conversion"
        customContentSpace="Model Name"
        bNoDefaultParam="0" bPostProcess="0"
  />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="observer conversion"
    customContentSpace="Pose Name"
    bNoDefaultParam="1" bPostProcess="0"

  />

  <param0 X="1.0" Y="0.0" Z="1.0" W="0.0" />
  <!-- X => scale X    Y => offset X    Z => scale Y    W => offset Y -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="sd" name="Display 1" />
  <display tDevice="sd" name="Display 2" />

  <param tConvert="exclude compound display" bNoDefaultParam="0" />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="camera content space translation"
    customContentSpace="Model Name"
    bNoDefaultParam="0" />

</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_1" />
  <display tDevice="dc" name="DC_2" />

  <param tConvert="merge calibration" />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="auto frustum"
    bNoDefaultParam="1"

  />

  <eyePoint X="100.0" Y="100.0" Z="100.0" />
  <param0 X="0.0" Y="0.0" Z="0.0" W="1.777" />
  <!-- X => unused    Y => unused    Z => unused    W => preferred ratio -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="clone calibration" />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="remove P2C outlier"
    bNoDefaultParam="1"
  />
  <param1 X="8.0" Y="0" Z="0.0" W="0.0" />
  <!-- X => filter size    Y => method    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="smooth P2C"
    bNoDefaultParam="1"
  />
  <param1 X="8.0" Y="0" Z="0.0" W="0.0" />
  <!-- X => filter size    Y => method    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="smooth P2C"
    bNoDefaultParam="1"
  />
  <param1 X="8.0" Y="0" Z="0.0" W="0.0" />
  <!-- X => filter size    Y => method    Z => unused    W => unused -->
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="PrjSpace overlap" />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="Crop Display Geometry"
    bNoDefaultParam="1"
    path="c:\Mask.png"
  />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="Crop Display Geometry with Projector Mask" />
</define>

```

```

<define name="Convert" type="CalibCommerce" >
  <display tDevice="dc" name="DC_Final" />
  <param tConvert="delete calibration" />
</define>

```

3.4 action="Configure" state="ExportConfig"

Typ des separaten Parameterblocks: CalibCommerce

Verfügbare Sektionen:

- es können mehrere <display> Sektionen angegeben werden
- bis auf tConvert sind alle Attribute in <param> Sektion optional

Sektion	Attribut	Wert	Beschreibung
<param>	tConvert	vwf separated	ein vwf-/ bmp-Dateipaar pro exportierter Display-Kalibrierung
		vwf	Eine vwf-Datei pro exportiertem Display-Compound Kalibrierung
		DWM	nicht dokumentiert
		x-file	Kalibrierungen werden im x-Dateiformat exportiert
		OpenWarp	Kalibrierungen im OpenWarp-Dateiformat exportiert
		MIPS	Kalibrierungen werden im Barco MIPS kompatiblen Dateiformat

			exportiert
		MPCDI v1	Kalibrierungen werden im MPCDI Version 1 Format exportiert
		MPCDI	Kalibrierungen werden im MPCDI Version 2 Format exportiert
		SGC	Kalibrierungen werden im Sciss SGC Format exportiert
		openEXR	Kalibrierungen werden in ILM's openEXR Format exportiert
		DISI	Kalibrierungen werden im Rheinmetall DISI X-treme Format exportiert
		NLIM	Kalibrierungen werden im IG NLIM Format exportiert
		CalibData	Kalibrierungen werden im „Calibration Data File“ Format exportiert
	path	[Pfadangabe]	Definiert den Ordner in dem die erzeugten Dateien abgespeichert werden sollen
	name	[Beschreibung/ Dateiname]	Definiert den Dateiprefix bzw. den exakten Dateinamen wenn bExactFileName="1"
	bNoVC	[0, 1]	Kalibrierungen sollen ohne Beachtung der virtuell canvas exportiert werden
	bNoGeom	[0, 1]	die geometrische Kalibrierung soll nicht exportiert werden
	bNoBlend	[0, 1]	das blending der exportierten Displays soll nicht exportiert werden
	bNoMask	[0, 1]	Die Kalibrierungen sollen ohne eventuell gesetzte Masken exportiert werden
	bNoBlack	[0, 1]	Die Kalibrierungen sollen ohne Blacklevel-Kompensation exportiert werden
	bSplitDisplays	[0, 1]	Die einzelnen Teile eines gesplitteten Displays sollen separat exportiert werden
	bExactFileName	[0, 1]	name Attribut gibt den exakten Dateinamen an (nur verfügbar für tConvert="vwf")
	bScaleSplitDisplays	[0, 1]	Teile eines gesplitteten Displays

			werden auf das gesamte Display skaliert
	b3D	[0, 1]	Statt des Inhaltsmappings werden die 3D Informationen der Displays exportiert
	bToMaster	[0, 1]	Wird eine multi-client Kalibrierung exportiert, werden die Exports aller Clients auf dem ausführenden Rechner gespeichert
	bVirtualContentRect	[0, 1]	Anwendung eines virtuellen Inhaltsrechtecks
	bBlankUnused	[0, 1]	Unbenutzte Teile eines gesplitteten Displays werden geschwärzt, ansonsten wird ein homogenes Inhaltsmapping exportiert
	bExtendedDefFile	[0, 1]	Verwendung einer Konfigurationsdatei zur Durchführung komplexer Exportaufgaben
	bSilent	[0, 1]	Ausführung der Task ohne visuelle Ausgabe
	bNoAutoFileCreation	[0, 1]	Unterdrückt die automatische Erzeugung von Export-Dateien
	bUseBasePath	[0, 1]	Exportiert in den Root Datenpfad anstatt in den Export Unterordner (wenn kein expliziter Pfad angegeben wurde)
	bNoDefaultParam	[0, 1]	Unterbindet die Verwendung von Standard-Parametern. Alle benötigten Parameter müssen angegeben werden.
	bPostProcess	[0, 1]	Aktiviert/Deaktiviert das filtern von Berechnungsergebnissen
	bMergeRemote	[0, 1]	Wird eine multi-client Kalibrierung exportiert, werden alle Exports für alle Clients in eine Export-Dateien geschrieben.
	bFixParam	[0, 1]	Wenn aktiviert, sollten alle zusätzlichen Parameter in der UI deaktiviert werden.
	gammaBlend	[0.0 .. 10.0]	Erlaubt die Definition eines Gammawertes, welcher auf die Blending-Texture angewendet wird (0.0 für deaktiviert).
<grid>	row	[1 .. n]	Gibt für nicht pixelbasierte

			Exportformate die Anzahl der Zeilen an
	col	[1 .. n]	Gibt für nicht pixelbasierte Exportformate die Anzahl der Zeilen an
<display>	tDevice	sd	einzelnes Display
		dc	Display Compound
		sc	Super Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds
<source>	tSource	Core	Der SPeASY Kern ist Quelle (id Parameter muß gesetzt sein)
		GUI	Die GUI ist Quelle
		Network	Das Netzwerk ist Quelle (id und idString Parameter muß gesetzt sein)
		Process	Ein anderer Prozess ist Quelle
		ProjectDesign	Ein Projekt-Design ist Quelle (idString Parameter muß gesetzt sein)
	id	[-1 .. n]	Optionale ID der Quelle
	idString	[Bezeichner]	Optionaler Identifikator der Quelle

```

<define name="Export" typ="CalibCommerce" >
  <display tDevice="sd" name="D2 SyncMaster (SAM04EA)" />
  <param tConvert="vwf"
    path="C:\Export"          name="Script_Test.vwf"
    bNoVC="0"                 bNoGeom="0"
    bNoBlend="0"              bNoMask="0"
    bNoBlack="0"
    bSplitDisplays="0"        bExactFileName="1"
    bScaleSplitDisplays="0"   b3D="0"
    bToMaster="1"             bVirtualContentRect="0"
    bBlankUnused="0"          bExtendedDefFile="0"
    bNoDefaultParam="0"       bPostProcess="1"
    gammaBlend="0.0"
  />
  <grid row="21" col="21" />
</define>

```

3.5 action="Configure" state="Validate"

Typ des separaten Parameterblocks: **RenderPipe**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	extern	[Variable]	Referenz auf eine ini-Datei Variable mit benötigten Parametern

```
<define name="RP1.config" typ="RenderPipe" >
  <param extern="$ARVA$Assign$View1$" />
</define>
```

oder

Sektion	Attribut	Wert	Beschreibung
<source>	tDevice	sd	capturen des Inhaltes eines Display
		file	anzeigen des Inhalts einer Datei
		camera	anzeigen der Ausgabe einer Kamera/Live-Input Gerätes
	name	[Bezeichner]	Name des Gerätes wenn tDevice="sd" oder tDevice="camera" , wenn kein name bzw. iDeviceGroup angegeben wird, wird die erste Entität verwendet
	iDeviceGroup	[Index]	wenn kein name und tDevice nicht " file ", bestimmt den Index (beginnend mit 1) des zu verwendenden Gerätes innerhalb des Gerätetyps
	file	[Dateiname]	Name der Datei wenn tDevice="file" (optional)
<target>	tDevice	sd	einzelnes Display
		dc	Display Compound
		sc	Super Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds, wenn kein name bzw. iDeviceGroup

			angegeben wird, wird die erste Entität verwendet
	iDeviceGroup	[Index]	wenn kein name angegeben, bestimmt den Index (beginnend mit 1) des zu verwendenden Gerätes innerhalb des Gerätetyps
	ip	[IP]	IP-Adresse eines PC, wenn tDevice ein Ziel auf einem anderen PC beschreibt (optional)
<param>	bUseVC	[0, 1]	Aktiviert/Deaktiviert die Verwendung der virtual canvas
	bUseGeomCorr	[0, 1]	Aktiviert/Deaktiviert die Verwendung der geometrischen Korrektur
	bUseColorCorr	[0, 1]	Aktiviert/Deaktiviert die Verwendung der Untergrund Farbkorrektur (wenn verfügbar)
	bAABorderSmooth	[0, 1]	Aktiviert/Deaktiviert die Verwendung der Randweichzeichnung
	bUseEdgeBlend	[0, 1]	Aktiviert/Deaktiviert die Verwendung des edge blendings
	bAutoSave	[0, 1]	Aktiviert/Deaktiviert die automatische Speicheroption
	bCompleteInit	[0, 1]	Aktiviert/Deaktiviert die Verwendung der vollständigen Initialisierungsoption bei Start
	bAugmentedElements	[0, 1]	Aktiviert/Deaktiviert die Anzeig von erweiterten Inhalten

```

<define name="RP1.config" typ="RenderPipe" >
  <source tDevice="file" file="E:\test.bmp" />
  <target tDevice="sd" name="D1 (CMN1343)" ip="192.168.100.5" />
  <param bUseVC="1" bUseGeomCorr="1" bUseColorCorr="0"
    bAABorderSmooth="0"
    bUseEdgeBlend="1"
    bAutoSave="0" bCompleteInit="0" bAugmentedElements="0"
  />
</define>

```

3.6 action="Configure" state="StartOption"

Typ des separaten Parameterblocks: **RenderPipe**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	bUseContentCache	[0, 1]	Aktiviert/ Deaktiviert die Verwendung des Content Cache

```
<define name="RP1.start" typ="RenderPipe" >
  <param bUseContentCache="1" />
</define>
```

3.7 action="Configure" state="all"

3.7.1 type="Renderer" subtype="dx9"

Typ des separaten Parameterblocks: **common**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname/Variable]	Name der mrd-Datei, die zur Konfiguration des Renderers verwendet werden soll

```
<define name="Renderer.Config" typ="Common" >
  <param file="Test.mrd" />
</define>
```

3.8 action="Create" typ="Behaviour"

Typ des separaten Parameterblocks: **BehaviourCreate**

In der Regel ist dieser Parameterblock nicht nötig.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	interactLevel	many	vollständige Interaktion
		reduced	reduzierte Interaktion

		less	minimale Interaktion
		minimal	möglichst keine Interaktion

```
<define name="Cl.Start" typ="BehaviourCreate" >
  <param interactLevel="many"
  />
</define>
```

3.9 action="Create" typ="Behaviour" subtype="SingleClientCalib"

Typ des separaten Parameterblocks: BehaviourCreate

Verfügbare Sektionen:

- es können mehrere <display> Sektionen angegeben werden

Sektion	Attribut	Wert	Beschreibung
<param>	interactLevel	many	vollständige Interaktion
		reduced	reduzierte Interaktion
		less	minimale Interaktion
		minimal	möglichst keine Interaktion
		folgende Werte können zusätzlich durch Komma getrennt angegeben werden	
		NoInterResult	Das Ergebnis der einzelnen Display Kalibrierungen soll nicht angezeigt werden.
		NoFinalResult	Das Resultat aller kalibrierten Displays soll nicht angezeigt werden.
<display>	tDevice	sd	einzelnes Display
		dc	Display Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds
	iDeviceGroup	[Index]	Wenn kein name angegeben, bestimmt den Index (beginnend mit 1) des zu verwendenden Gerätes innerhalb des Gerätetyps

```
<define name="Cl.Start" typ="BehaviourCreate" >
  <param interactLevel="many, noInterResult"
  />
  <display tDevice="sd" name="D1 NEC LCD1760NX (NEC6604)" />
</define>
```

3.10 action="Create" typ="CntSpace" subtyp="3DModel"

Typ des separaten Parameterblocks: CntSpace

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	modelFile	[Dateiname/Variable]	definiert die 3D-Modell Datei
	matrixFile	[Dateiname/Variable]	optional, definiert eine XML-Datei mit einer Transformationsmatrix für das verwendete 3D-Modell
	matrixNode	[Bezeichner]	optional, definiert die Sektion/Attribut in der die Transformationsmatrix in matrixFile definiert ist
	cullMode	cw	Verdeckungsmodus Uhrzeigersinn
		ccw	Verdeckungsmodus gegen den Uhrzeigersinn
		none	kein Verdeckungsmodus
	cntSpaceW	[1 .. n]	bezeichnet die Größe des verwendeten Inhaltsraum in x Richtung
	cntSpaceH	[1 .. n]	bezeichnet die Größe des verwendeten Inhaltsraum in y Richtung
	bVolatileModel	[0, 1]	zeigt an, daß die 3D-Modell Datei vor jeder Berechnung neu eingelesen werden soll
	bSuppressRootTrans	[0, 1]	unterdrückt eine eventuell in der 3D-Modell Datei enthaltene Basis-Transformationsmatrix
	bTransposeTransMat	[0, 1]	definiert, daß eine optional verwendete Transformationsmatrix (matrixFile) vor der Verwendung transponiert werden muß

```

<define name="model param" typ="CntSpace" >
  <param modelFile="D:\model.obj"
    matrixFile="D:\model.matx" matrixNode="Pose/Matrix4x4"
    cullMode="none"
    cntSpaceW="3072"           cntSpaceH="3072"
    bVolatileModel="1"         bSuppressRootTrans="1"
    bTransposeTransMat="1"
  />
</define>

```

3.11 action="Create" typ="file" subtyp="bmp"

Typ des separaten Parameterblocks: CreateImage

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	tImage	RGB	erzeugt ein 24bit RGB Bild
		RGBA	erzeugt ein 32bit RGBA Bild
		GRAY	erzeugt ein 8bit Graustufen Bild
	width	[Größe]	definiert die Größe des Bildes in x Richtung
	height	[Größe]	definiert die Größe des Bildes in y Richtung
<background>	red	[Farbwert]	[0 .. 255] definiert den Rotwert der Hintergrundfarbe
	green	[Farbwert]	[0 .. 255] definiert den Grünwert der Hintergrundfarbe
	blue	[Farbwert]	[0 .. 255] definiert den Blauwert der Hintergrundfarbe
	alpha	[Alphawert]	[0 .. 255] wenn tImage="RGBA", definiert den Alphawert der Hintergrundfarbe
<target>	file	[Dateiname]	bestimmt den Namen der zu erzeugenden Datei

```
<define name="CntFile" typ="CreateImage" >
  <param tImage="RGBA" width="800" height="600" />
  <background red="255" green="255" blue="0" alpha="255" />
  <target file="Cnt.bmp" />
</define>
```

3.12 action="Execute" typ="Timer" subtyp="Sleep"

Typ des separaten Parameterblocks: Common

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	duration	[1 .. n]	Dauer in Millisekunden


```
<define name="StdWait" typ="Common" >
  <param duration="2000" />
</define>
```

3.13 action="Execute" typ="App"

Typ des separaten Parameterblocks: **App**

Anwendbar für subtyp: **start**, **restart**, **shutdown**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	tmSleep	[1 .. n]	Dauer in Millisekunden bevor die Aktion ausgeführt wird
	name	[Dateiname]	Dient zur Angabe des Dateinamens der Anwendung auf die sich die Aktion bezieht. Wenn name nicht angegeben wurde oder leer ist, beziehen sich die Parameter auf die ausführende Anwendung.
	param	[String]	kann verwendet werden um Kommandozeilen-Parameter zu definieren
	bWaitUntilFinished	[0 .. 1]	Für subtyp: start , ermöglicht das warten auf das Beenden der gestarteten App

```
<define name="CloseApp" typ="App" >
  <param name="notepad.exe" bWaitUntilFinished="1" />
</define>
```

```
<define name="Restart" typ="App" >
  <param tmSleep="3000" param='/H' />
</define>
```

3.14 action="Execute" typ="OS"

Typ des separaten Parameterblocks: **OS**

Anwendbar für subtyp: **restart**, **shutdown**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	tmShowInfo	[1 .. n]	Definiert die Anzeigedauer in Sekunden eines Informationsdialogs, wenn info angegeben wurde
	info	[String]	Kann verwendet werden um einen Hinweistext zu definieren, der bei Ausführung der Aktion angezeigt wird
	bForceAppClose	[0 .. 1]	Wenn aktiviert, werden laufende Anwendungen automatisch beendet (es können Daten verloren gehen)

```

<define name="Restart" typ="OS" >
  <param tmShowInfo="10" info="System restart" bForceAppClose="1" />
</define>

```

3.15 **action="Load", "Save" typ="file" subtyp="sps"**

Typ des separaten Parameterblocks: **Common**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname]	Name der sps-Kalibrierungsdatei, die geladen/ speichert werden soll
	bMustNotExist	[0 .. 1]	(optional) verfügbar für action="Load". Steuert ob das Skript weiter ausgeführt wird, wenn die Datei nicht existiert
	fParam	[0 ..]	(optional) kann verwendet werden um zu bestimmen, welche Informationen aus der Datei geladen werden sollen. Additiv als Summe sind folgende Werte zulässig:
		1	Die Display-Kalibrierungen sollen geladen werden
		2	Der Präsentationszustand der Ausgabe-Linien soll geladen werden
		8	Die individuellen Resource Parameter sollen geladen werden

		128	Die Display-Split Informationen sollen geladen werden
		32	Die Ausgabe-Linien sollen nicht automatisch starten
		256	Die Ausgabe-Linien sollen alle nach dem Laden gestartet werden
		512	Läd auch Kalibrierungen, die keinem Display zugeordnet werden können
<conditions>			verfügbar für action="Load". Ermöglicht das bedingte laden einer Datei Siehe Abschnitt 3.34

Wenn die Datei „Calib.sps“ existiert, dann wird diese nur geladen, wenn das Display-Compound mit dem Namen „screen0“ **nicht** existiert.

```
<define name="sps" typ="Common" >
  <param file="Calib.sps" bMustNotExist="1" />
  <conditions bAll="0" >
    <condition tCondition="calibnotexist" tDevice="dc" name="screen0" />
  </conditions>
</define>
```

oder

Sektion	Attribut	Wert	Beschreibung
<param>	sel	[Selektor/Variable]	verfügbar für action="Load" definiert einen Selektor, mit welcher der gewünschte Dateiname aus einer Liste ausgewählt wird
	list	[Name]	verfügbar für action="Load" definiert den Namen der Liste, in der gesucht werden soll
	bOnlyNew	[0, 1]	verfügbar für action="Load" wenn aktiviert, die "Load" Aktion soll nur ausgeführt werden, wenn eine neue Selektion erfolgt

3.16 *action="Load", "Save" typ="file" subtyp="vc"*

Typ des separaten Parameterblocks: **Common**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname]	Name der vc-Definitionssdatei, die geladen/ speichert werden soll
	bMustNotExist	[0 .. 1]	(optional) verfügbar für action="Load". Steuert ob das Skript weiter ausgeführt wird, wenn die Datei nicht existiert
<display>	tDevice	sd	einzelnes Display
		dc	Display Compound
		sc	Super Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds für das die virtuell canvas geladen/ gespeichert werden soll

```
<define name="canvas" typ="Common" >
  <param file="Calib.vvc" />
  <display tDevice="dc" name="Compound 187" />
</define>
```

3.17 *action="Load", "Save" typ="file" subtyp="playlist"*

Typ des separaten Parameterblocks: **Common**

! Das **name** Attribut der Task spezifiziert die Presenterlinie, für welche die Playlist geladen/ gespeichert werden soll !

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname]	Name der Playlist Datei
	bMustNotExist	[0 .. 1]	(optional) verfügbar für action="Load". Steuert ob das Skript weiter ausgeführt wird, wenn die Datei nicht existiert

```
<define name="Playlist" typ="Common" >
  <param file="LastPlaylist.xml" />
</define>
```

3.18 **action="Save" typ="file" subtyp="calibuserdata"**

Typ des separaten Parameterblocks: **Common**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname]	Name der Zieldatei
<display>	tDevice	dc	Display Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds für das die virtuell canvas geladen/ gespeichert werden soll

```
<define name="userdata" typ="Common" >
  <param file="CalibData.dat" />
  <display tDevice="dc" name="Final" />
</define>
```

3.19 **action="Load" typ="file" subtyp="calibuserdata"**

Typ des separaten Parameterblocks: **Common**

(Die Informationen in der Quelldatei werden über den Namen der Kalibrierung zugeordnet.)

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	file	[Dateiname]	Name der Quelldatei
	bMustNotExist	[0 .. 1]	(optional) Steuert ob das Skript weiter ausgeführt wird, wenn die Datei nicht existiert

```

<define name="userdata" typ="Common" >
  <param file="CalibData.dat" bMustNotExist="1" />
</define>

```

3.20 *action="Transfer" typ="file"*

Typ des separaten Parameterblocks: **Transfer**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<from>	file	[Dateiname]	Name der Datei, die transferiert (kopiert) werden soll
<to>	file	[Dateiname]	Name der Zieldatei (Datei wird überschrieben falls sie bereits existiert)

```

<define name="Backup" typ="Transfer" >
  <from file="SH_Compounds.sps" />
  <to   file="E:\Backup\SH_Compounds.sps" />
</define>

```

3.21 *action="Transfer" typ="Export"*

Typ des separaten Parameterblocks: **Transfer**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<source>	tSource	Core	Der SPeASY Kern ist Quelle (id Parameter muß gesetzt sein)
		GUI	Die GUI ist Quelle
		Network	Das Netzwerk ist Quelle (id und idString Parameter muß gesetzt sein)
		Process	Ein anderer Prozess ist Quelle
		ProjectDesign	Ein Projekt-Design ist Quelle (idString Parameter muß gesetzt sein)
	id	[-1 .. n]	Optionale ID der Quelle

	idString	[Bezeichner]	Optionaler Identifikator der Quelle
<local>/ <remote>	bAll	[0 .. 1]	Gibt an, das die Exportdateien aller PCs transferiert werden sollen
	bBackup	[0 .. 1]	Gibt an, das eine Backup aller Dateien, die überschrieben werden, angelegt werden soll. Nur verfügbar wenn backupSuffix angegebenen wurde.
	path	[Pfadangabe]	Definiert den Ordner in dem die erzeugten Dateien abgespeichert werden sollen
	fileFmt	[Dateiformat]	Definiert den Dateinamen (ohne Erweiterung!) der erzeugten Dateien. Erlaubt sind Platzhalter: - <i>\$ip\$</i> : IP-Adresse - <i>\$org\$</i> : original Dateiname
	backupSuffix	[Bezeichner]	Definiert eine Erweiterung, mit der Backup-Dateien markiert werden
	core	[IP]	Verfügbar für <remote> Spezifiziert die IP-Adresse des gewünschten PCs, leere Angabe als Platzhalter für alle Remote PCs
	tmOut	[0 .. n]	Verfügbar für <remote> Antwort Wartezeit in Millisekunden

```

<define name="Backup" typ="Transfer" >
  <source tSource="ProjectDesign"
    idString="8E113541-5698-488A-9FDF-5888F088AB22" />
  <local bAll="1" bBackup="1" path="e:\test" fileFmt="_Startup_$ip$"
    backupSuffix="bkp" />
  <remote core="" bAll="0" bBackup="1" path="e:\test_remote"
    fileFmt="_Startup_$ip$" backupSuffix="bkp" tmOut="5000" />
</define>

```

Transferiert die Export-Dateien einer Export-Task, bezogen auf spezifiziertes Projekt-Design. Speichert lokal alle Export-Dateien für alle PCs im Ordner „[E:\test](#)“ und sichert vorher alle zu überschreibenen Dateien durch anhängen der Erweiterung „bkp“.

Weiterhin werden die spezifischen Export-Dateien für beteiligte remote PCs auf diesen im Ordner „[E:\test_remote](#)“ gespeichert und auch dort vorherige Dateien gesichert.

3.22 action="Playlist"

Typ des separaten Parameterblocks: **Playlist**

Die Liste von angegebenen <itemchange> Sektionen werden von oben nach unten abgearbeitet.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<itemchange>	tOp	add	Fügt ein neues Item der Playlist hinzu
		delete	Löscht ein Item/ alle Items aus der Playlist
		refresh	aktualisiert ein existierendes Item/ alle Items der Playlist
	item	all	bezeichnet alle existierende Items (verfügbar für tOp ="delete" und tOp ="refresh")
		[Index]	bezeichnet das Item an Position Index in der Playlist (verfügbar für tOp ="delete" und tOp ="refresh")
		file	bezeichnet eine Mediendatei (verfügbar für tOp ="add")
		imgseq	bezeichnet eine Bildsequenz (verfügbar für tOp ="add")
		mrd	bezeichnet eine 3D-Modell Rendering Definition (verfügbar für tOp ="add")
		goto	bezeichnet ein „Gehe zu Item“ Steuer-Item (verfügbar für tOp ="add")
		stop	bezeichnet ein Stop Steuer-Item (verfügbar für tOp ="add")
		pause	bezeichnet ein Pause Steuer-Item (verfügbar für tOp ="add")
	param	[Bezeichner]	bestimmt eine Datei/ Dateifilter (verfügbar für tOp ="add" und item ="file, imgseq, mrd")
	flag	[CSV]	Komma separierte Liste mit zusätzlichen Parametern (verfügbar für tOp ="add")
		none	keine zusätzlichen Flags
		disabled	initial deaktiviertes Item

		autoplay	setzt das autoplay-Flag des Items
		mute	deaktiviert den Ton des Items
		select	wählt das Item automatisch aus
		infinite	signalisiert, daß das Item unendlich abgespielt werden soll (Bild Items)
		loop	signalisiert, daß das Item nach dem abspielen wiederholt abgespielt wird (Video, Bildsequenz Items)
		singleplay	kombiniertes Flag, setzt: loop, select, autoplay, infinite
	iPos	[Index]	verfügbar für tOp="add" bestimmt die Position, an der das neue Item eingefügt werden soll

```
<define name="RP.PL" typ="Playlist" >
  <itemchange tOp="delete" item="all" />
  <itemchange tOp="add" item="file" param="Cnt.bmp" flag="singleplay"/>
</define>
```

Löscht alle Items aus der Playlist. Fügt danach die Datei „Cnt.bmp“ (lokalisiert im Standard Content Ordner) mit „single play“ Eigenschaften der Playlist hinzu.

```
<define name="RP.PL" typ="Playlist" >
  <itemchange tOp="delete" item="6" />
</define>
```

Löscht das Item an Playlist Position 6.

3.23 **action="Render"**

Typ des separaten Parameterblocks: **RenderTarget**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<param>	tTarget	file	Das Ergebnis des Renderingschritts soll in eine Datei gespeichert werden.
	file	[Dateiname]	Name der Datei, wenn tTarget="file"

```
<define name="Renderer.Target" type="RenderTarget" >
  <param tTarget="file" file="Output.bmp" />
</define>
```

3.24 action="Manipulate" typ="Calibration" subtyp="Blending"

Typ des separaten Parameterblocks: BlendParam

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<display>	tDevice	dc	Derzeit werden nur Display-Compound Kalibrierungen unterstützt
	name	[Bezeichner]	Name des Display-Compounds
<param>	tMethod	uniform	setzen eines uniformen edge blending, keine zusätzlichen Parameter nötig
		standard	Standard edge blending Methode, benötigte Attribute: plateau, gradient, gamma, gammaPrj
		spline	Spline basierte edge blending Methode, benötigte Attribute: blackPt, whitePt, midPt, curvation, tensor, gamma, gammaPrj
	plateau	[0 .. 100]	Plateau-Parameter wenn tMethod="Standard"
	gradient	[-120 .. 120]	Gradient-Parameter wenn tMethod="Standard"
	gamma	[-100 .. 100]	Gamma-Parameter wenn tMethod="Standard" oder tMethod="Spline"
	gammaPrj	[2 .. 50]	Display Gamma-Parameter wenn tMethod="Standard" oder tMethod="Spline" (22 entspricht 2.2, 5 entspricht 0.5)
	blackPt	[0 .. 100]	Schwarz-Punkt Parameter wenn tMethod="Spline"
	whitePt	[0 .. 100]	Weiß-Punkt Parameter wenn tMethod="Spline"
	midPt	[0 .. 100]	Mittel-Punkt Parameter wenn tMethod="Spline"
	curvation	[0 .. 100]	Kurvigkeit Parameter wenn tMethod="Spline"
	tensor	[0 .. 100]	Tensor Parameter wenn tMethod="Spline"

```

<define name="SetBlend" type="BlendParam" >
  <display tDevice="dc" name="DC_Final" />
  <param tMethod="standard"
    plateau="50" gradient="8" gamma="0" gammaPrj="22"

  />
</define>

```

3.25 action="**Manipulate**" typ="**Calibration**" subtyp="**CalibChange**"

Typ des separaten Parameterblocks: **CalibChange**

Die Liste von angegebenen **<change>** Sektionen werden von oben nach unten abgearbeitet.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<display>	tDevice	dc	Derzeit werden nur Display-Compound Kalibrierungen unterstützt
	name	[Bezeichner]	Name des Display-Compounds
<param>	bUseVC	[0, 1]	Aktiviert/Deaktiviert die Verwendung der virtual canvas
	bUseGeomCorr	[0, 1]	Aktiviert/Deaktiviert die Verwendung der geometrischen Korrektur
	bUseColorCorr	[0, 1]	Aktiviert/Deaktiviert die Verwendung der Untergrund Farbkorrektur (wenn verfügbar)
	bAABorderSmooth	[0, 1]	Aktiviert/Deaktiviert die Verwendung der Randweichzeichnung
	bUseEdgeBlend	[0, 1]	Aktiviert/Deaktiviert die Verwendung des edge blendings
	bAutoSave	[0, 1]	Aktiviert/Deaktiviert die automatische Speicheroption
	bCompleteInit	[0, 1]	Aktiviert/Deaktiviert die Verwendung der vollständigen Initialisierungsoption bei Start
<change>	tManipulate	power value	Setzt die Leistungsparameter

			für alle Displays des Compounds bzw. für das spezifizierte Display. Benötigte Attribute: red , green , blue
		reset mask	Löscht die Masken aller Displays des Compounds bzw. für das spezifizierte Display.
		set mask	Setzt die Maske aller Displays des Compounds bzw. für das spezifizierte Display aus einer Bilddatei. Benötigte Attribute: maskFile
	tDevice	sd	Derzeit werden nur Einzel-Displays unterstützt. Werden tDevice und name nicht angegeben, wird die Aktion auf alle Displays des Compounds angewendet.
	name	[Bezeichner]	Name des Displays
	red	[0.0 .. 1.0]	Bestimmt den Leistungsparameter für den roten Farbkanal, wenn tMethod = “power value“
	green	[0.0 .. 1.0]	Bestimmt den Leistungsparameter für den grünen Farbkanal, wenn tMethod = “power value“
	blue	[0.0 .. 1.0]	Bestimmt den Leistungsparameter für den blauen Farbkanal, wenn tMethod = “power value“
	maskFile	[Dateiname]	Bilddatei, benutzt als Maske, wenn tMethod = “set mask“

```

<define name="ChangeTest" type="CalibChange" >
  <display tDevice="dc" name="DC_Final" />
  <param bUseVC="1" bUseGeomCorr="1" bUseColorCorr="0"
    bAABorderSmooth="0"
    bUseEdgeBlend="1"
    bAutoSave="0" bCompleteInit="0"
  />
  <change tManipulate="power value" tDevice="sd" name="Display 1"
    red="1.0" green="0.0" blue="0.0" />
  <change tManipulate="power value" tDevice="sd" name="Display 2"
    red="0.0" green="1.0" blue="0.0" />
  <change tManipulate="set mask" maskFile="DisplayMask.bmp" />
</define>

```

Diese define-Sektion aktiviert: VC, geometrische Korrektur und edge blending. Außerdem werden Oberflächen Farbkorrektur, antialiasing Rand, Randweichzeichnung, Autospeicherung und vollständige Initialisierung bei Start für das Display-Compound „DC_Final“ deaktiviert. Außerdem werden die Leistungsparameter für Display „Display 1“ auf komplett rot und für Display „Display 2“ auf komplett grün gesetzt. Als Letztes wird die Datei „DisplayMask.bmp“ als Maske für alle im Display-Compound „DC_Final“ zusammengefaßten Displays gesetzt.

3.26 **action="Manipulate" typ="Calibration" subtype="Mask"**

Typ des separaten Parameterblocks: **MaskParam**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<display>	tDevice	sd	Derzeit werden nur Einzel-Display-Kalibrierungen unterstützt
	name	[Bezeichner]	Name des Displays
<param>	scaleRed	[0 .. 100]	Skalierungsfaktor für den roten Farbkanal
	scaleGreen	[0 .. 100]	Skalierungsfaktor für den grünen Farbkanal
	scaleBlue	[0 .. 100]	Skalierungsfaktor für den blauen Farbkanal
	scaleInverseRed	[-100 .. 100]	Skalierungsfaktor für invertierten roten Farbkanal
	scaleInverseGreen	[-100 .. 100]	Skalierungsfaktor für invertierten grünen Farbkanal
	scaleInverseBlue	[-100 .. 100]	Skalierungsfaktor für invertierten blauen Farbkanal
	blackPt	[0 .. 100]	Gibt den Schwarz-Punkt an

	gamma	[-100 .. 100]	Spezifiziert einen Gamma-Wert.
	bAdjustColor	[0, 1]	Gibt an, ob die Farbe der Maske manipuliert werden soll.
	bUseInverseColor	[0, 1]	Gibt an, das die Farbkanäle der Maske invertiert werden sollen.
	bUseAlternativeGamma	[0, 1]	Aktiviert eine alternative Gamma-Methode
	bClampColor	[0, 1]	Gibt an, das die Farbkanäle zwischen (0 .. 1) begrenzt werden sollen.
	maskFile	[Dateiname]	Bilddatei, welche als Maske verwendet werden soll. Wenn dieses Attribut leer ist, wird die Maske auf Weiß gesetzt.

```

<define name="SetMask" type="MaskParam" >
  <display tDevice="sd" name="D1 PROJECTOR (DPC7401)" />
  <param scaleRed="100" scaleGreen="100" scaleBlue="100"
    scaleInverseRed="0" scaleInverseGreen="0" scaleInverseBlue="0"
    blackPt="0" gamma="0"
    bContentAligned="0"
    bAdjustColor="1" bUseInverseColor="0" bUseAlternativeGamma="0"
    bClampColor="1" maskFile="mask.bmp"

  />
</define>

```

3.27 **action="Manipulate" typ="Calibration" subtype="Blacklevel"**

Typ des separaten Parameterblocks: **BlacklevelParam**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<display>	tDevice	dc	Derzeit werden nur Display-Compound Kalibrierungen unterstützt
	name	[Bezeichner]	Name des Display-Compounds
<param>	value	[0 .. 100]	Blacklevel Korrektur Parameter
	maintainDark	[0 .. 100]	Steuert in wie weit dunkle Farbwerte erhalten werden

			sollen
	maintainBright	[0 .. 100]	Steuert in wie weit helle Farbwerte erhalten werden sollen

```

<define name="SetBlacklevel" type="BlacklevelParam" >
  <display tDevice="dc" name="DC_Final" />
  <param value="20" maintainDark="0" maintainBright="0"
  />
</define>

```

3.28 action="Manipulate" typ="ProjectManager"

Typ des separaten Parameterblocks: **ProjectManager**

Es können mehrere Operationen hintereinander angegeben werden.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<operation>	tOperation	start design	Startet den Projekt-Designer
		start activate	Expliziter Start des „Activate“ Schritts
	bQueued	[0,1]	Steuert, ob die Operation anschließend ausgeführt werden soll, wenn der Projekt-Manager aktuell eine Operation ausführt. Ansonsten wird die Operation abgelehnt.

```

<define name="Start_Designer" type="ProjectManager" >
  <operation tOperation="start design"/>
</define>

```

Startet den Projekt-Designer.

```

<define name="Start_Designer" type="ProjectManager" >
  <operation tOperation="start activate" bQueued="1" />
</define>

```

Startet explizit den „Activate“ Schritt. Wenn der Projekt-Manager aktuell eine Operation ausführt, wird die Operation unmittelbar im Anschluß ausgeführt.

3.29 action="**Manipulate**" typ="**NativeWB**"

Typ des separaten Parameterblocks: **NativeWB**

Es können mehrere Operationen hintereinander angegeben werden.

Angewendet werden die Operationen auf Displays, welche vormals exportiert und über **<source>** adressiert werden.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<source>			Siehe Abschnitt 3.4
<operation>	tOperation	apply	Setzt native warp Feature wie warping, blending
		clear	Löscht native warp Feature wie warping, blending
		set overlap	Setzt ein natives Display-Overlap
		clear overlap	Löscht ein natives Display-Overlap
	tDevice	dc	Display Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds
	tDestination	local	Die Operation soll auf alle beteiligten lokalen Displays angewendet werden
		remote	Die Operation soll auf alle beteiligten Client Displays angewendet werden
	tmOutAccess	[0 .. n]	Bestimmt die Wartezeit in ms um Zugang zu internen Strukturen zu erhalten
	tmWait	[0 .. n]	Bestimmt eine feste Wartezeit in ms, ausgeführt nach bestimmten Treiber-Funktionen um Treiber-spezifische Probleme zu minimieren
<operation> <standard> <param>			Ermöglicht die Angabe von Display-spezifischen Parametern. Wurden keine Display-spezifischen Parameter angegeben, werden die <standard> Parameter verwendet.
	strOSDispID	[Bezeichner]	OS Display ID
	hwDisplayID	[ID]	Treiber spezifische Display ID

			Nummer. (wenigstens strOSDispID oder hwDisplayID muß angegeben werden)
	fFeature	warp	Das warp-Feature wird manipuliert
		blend	Das blending-Feature wird manipuliert
		simpleTriangulation	Verwendet eine vereinfachte Methode zur Erzeugung von warp-Meshs
		auto	Berechnet das Overlap, basierend auf Display-warp
		overlapFromSize	overlapH/ overlapV gibt die gewünschte Displayauflösung an (der tatsächliche wird berechnet)
	qTessellate	[0 .. n]	Bestimmt die Tessellation-Weite in Pixel zum erzeugen ausgedünnter warp-Meshs verfügbar für apply/clear warp
	corrOverlapX	[0 .. n]	Korrekturfaktor in x-Richtung, verwendet beim anwenden von warp (kompensiert Treiber-spezifische Fehler) verfügbar für apply/clear warp
	corrOverlapY	[0 .. n]	Korrekturfaktor in y-Richtung, verwendet beim anwenden von warp (kompensiert Treiber-spezifische Fehler) verfügbar für apply/clear warp
	overlapH	[0 .. n]	Bestimmt den horizontalen Overlap (abhängig von fFeature) verfügbar für set overlap/ clear overlap
	overlapV	[0 .. n]	Bestimmt den vertikalen Overlap (abhängig von fFeature) verfügbar für set overlap/ clear overlap

```

<define name="SetWarp"  type="NativeWB" >
  <source tSource="GUI" />
  <operation tOperation="apply" tDevice="dc" name="screen0"
    tDestination="local" tmOutAccess="10000" tmWait="3000">
    <standard fFeature="warp,blend" qTessellate="50"
      corrOverlapX="0" corrOverlapY="0"/>
    <param strOSDispID="\\.\DISPLAY2" hwDisplayID="2147881095"
      fFeature="warp,blend" qTessellate="100"
      corrOverlapX="0" corrOverlapY="0"/>
  </operation>
</define>

```

```

<define name="ClearAll" type="NativeWB" >
  <source tSource="GUI" />
  <operation tOperation="Clear Overlap" tDevice="dc" name="screen0"
    tDestination="local" tmOutAccess="10000" tmWait="3000">
    <standard fFeature=""/>
  </operation>
  <operation tOperation="Clear" tDevice="dc" name="screen0"
    tDestination="local" tmOutAccess="10000" tmWait="3000">
    <standard fFeature="warp, blend"/>
  </operation>
</define>

```

Löscht das warp/blend Feature und gesetztes Overlap für alle in Display-Compound „screen0“ adressierten lokalen Displays, exportiert durch den Nutzer via UI.

3.30 *action="Manipulate" typ="GUI"*

Typ des separaten Parameterblocks: **GUI**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<element>	tSector	main	Haupt-UI bzw. allgemeine Elemente
		presentation	Presenter-Linie Elemente
	tIdent	type	Das Element wird über seinen Typ adressiert
		id	Das Element wird über seine ID adressiert
	tElement	window	Window Element
		button	Knopf Element
		combo	Kombobox Element
		menu	Menü Element
	id	activate button	Aktivier-/Deaktivier Knopf
		calibrate button	Kalibrier-Knopf
		recalibrate button	Rekalibrier-Knopf
		playlist button	Playliste-Knopf
		liveinput button	Live-Input-Knopf
		colorblend button	Farbe/Blend-Knopf
		display compound button	Display-Compound-Knopf
		additional output button	Spezielle Ausgabe-Knopf
		presopt fullscreen button	Presenter-Option: Vollbild
		presopt stacking button	Presenter-Option: Blending

			Aus/An
		presopt auto save button	Presenter-Option: autom. Speichern (Netzwerk)
		presopt full init button	Presenter-Option: alles neu übertragen (Netzwerk)
		presopt color correction button	Presenter-Option: pixelgenaue Farbkorrektur
		mrd adjust dialog	Modell-Rendering Anpassung Dialog
		wait dialog	Warte-Info-Dialog
		main menu	Hauptfenster-Menü
		device target	Ausgabe-Ziel
		notify task finish	Spezielle Benachrichtigung über die Beendigung einer Task
		[id]	Wenn tIdent=id , die ID des Elements
	behaviour	[name]	Name des Verhaltens abhängig von tSector
<element>	tRequest	set window text	Verändert die Beschriftung eines Elements
		enable window	aktiviert/deaktiviert ein Element
		create client window	erzeugt ein Client-Fenster
		destroy client window	zerstört ein Client-Fenster
	fSemantic		durch Komma getrennte Zustandsangaben
		active	Beschreibt den aktiv-Zustand
		inactive	Beschreibt den inaktiv-Zustand
		initial	Beschreibt den initial-Zustand
	bEnable	[0,1]	Gibt an, ob angegebener Zustand aktiviert/deaktiviert werden soll

```

<define name="Enable_Activate" type="GUI" >
  <element tSector="presentation" tIdent="typ" id="activate button"
    behaviour="RP0"/>
  <param tRequest="enable window" bEnable="1" fSemantic="active"/>
</define>

```

Aktiviert den „Activate“-Knopf von Presenterlinie „RP0“ (durch aktivieren des „active“ Zustandes)

```

<define name="Enable_Playlist" type="GUI" >
  <element tSector="presentation" tIdent="typ" id="playlist button"
    behaviour="RP0"/>
  <param tRequest="enable window" bEnable="1" fSemantic="active, inactive"/>
</define>

```

Aktiviert den „Playlist“-Knopf von Presenterlinie „RP0“ (durch aktivieren des „active“ und „inaktiv“ Zustandes, dadurch entscheidet der Zustand der Presenterlinie ob der Knopf aktiv ist)

```

<define name="Disable_Playlist" type="GUI" >
  <element tSector="presentation" tIdent="typ" id="playlist button"
    behaviour="RP0"/>
  <param tRequest="enable window" bEnable="0" fSemantic="active, inactive"/>
</define>

```

Deaktiviert den „Playlist“-Knopf von Presenterlinie „RP0“ (durch deaktivieren des „active“ und „inaktiv“ Zustandes, dadurch ist der Knopf immer inaktiv, unabhängig vom Zustand der Presenterlinie)

```

<define name="Disable_Playlist" type="GUI" >
  <element tSector="main" tIdent="typ" id="wait dialog"/>
  <param tRequest="create client window"/>
</define>

```

Erzeugt einen Warte-Dialog (Dialog mit Fortschrittsbalken)

3.31 **action="Send" typ="TCP" subtype="Command"**

Typ des separaten Parameterblocks: **NetCommand**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<target>	core	[IP]	IP-Adresse des gewünschten PCs
	iPort	[1 .. n]	Portnummer
<param>	command	[String]	Kommando was gesendet werden soll (siehe Remote Control Reference Dokumentation) !! &-Zeichen als & kodieren !! Nicht druckbare Zeichen in der Form: "%hh" mit [hh] : 2stelliger Hexcode (Bsp.: "%0D%0A" => "\r\n")

	tmOut	[0 .. n]	Antwort Wartezeit in Millisekunden
	tFormat		Optional, das angegebene Kommando wird im angegebenen Format gesendet
		HTTP_GET	ein HTTP GET Request Header wird verwendet
		HTTP_POST	ein HTTP POST Request Header wird verwendet

```
<define name="test" type="NetCommand" >
  <target core="192.168.2.20" iPort="8080" />
  <param command="control.htm?present=1" tmOut="4000" />
</define>
```

Sendet ein einfaches (String) Kommando.

```
<define name="test" type="NetCommand" >
  <target core="192.168.2.20" iPort="8080" />
  <param command="GET /control.htm?present=1 HTTP/1.1%0D%0A%0D%0A"
    tmOut="4000" />
</define>
oder
<define name="test" type="NetCommand" >
  <target core="192.168.2.20" iPort="8080" />
  <param command="/control.htm?present=1" tmOut="4000" tFormat="HTTP_GET" />
</define>
```

Sendet das gleiche Kommando als HTTP-Request.

3.32 **action="Send" typ="TCP" subtyp="File"**

Typ des separaten Parameterblocks: **NetCommand**

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<target>	core	[IP]	IP-Adresse des gewünschten PCs
	iPort	[1 .. n]	Portnummer
<param>	source	[Dateiname]	Spezifiziert die Quelldatei (mit Pfadangabe wenn nicht in Standardordner)
	target	[Dateiname]	Spezifiziert den Namen der Zieldatei. Wenn nicht angegeben, wird der Quell-

			Dateiname verwendet. Wird kein Pfad angegeben, wird die Datei im jeweiligen Standardordner erzeugt.
	tFile	Content	Content (Bild) Datei
		Calibration	Kalibrierungsdatei
		Export	Export-Dateien
		3DModel	3D-Modell-Datei
		Script	Skript-Datei
		Execute	ausführbare Datei
		Playlist	Playlist-Datei
	tmOut	[0 .. n]	Antwort Wartezeit in Millisekunden
	command	[String]	Kommando was gesendet werden soll (siehe Remote Control Reference Dokumentation) !! &-Zeichen als &kodieren !! Wenn nicht angegeben, wird: "/file" verwendet.

```

<define name="test" type="NetCommand" >
  <target core="192.168.2.20" iPort="8080" />
  <param source="datei.png" target="new.png" tFile="content" tmOut="4000" />
</define>

```

3.33 **action="IPC" typ="Common"**

Typ des separaten Parameterblocks: **Common**

Es können mehrere **<perform>** Sektionen angegeben werden.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<perform>	tAction	finish	Beendet einen spezifischen (tTask) bzw. aktuellen IPC-Task
	tTask	guided	Spezifiziert einen geführten Kalibrierungs-/Manipulations-IPC-Task
	iError	[0 .. n]	Kann verwendet werden um eine Fehler-ID zu spezifizieren.

```

<define name="test" type="Common" >
  <perform tAction="finish" tTask="guided" iError="1" />
</define>

```

3.34 action="Condition"

Typ des separaten Parameterblocks: **Condition**

Es können mehrere **<option>** Sektionen angegebenen werden.

Verfügbare Sektionen:

Sektion	Attribut	Wert	Beschreibung
<maker>	tMaker	GUI	Spezifiziert, daß die Bedingung durch die GUI entschieden werden muß
		core	Spezifiziert, daß die Bedingung durch den Kern evaluiert wird
		network	Spezifiziert, daß die Bedingung durch einen Netzwerk-Request evaluiert wird
		process	Spezifiziert, daß die Bedingung durch einen anderen Prozess evaluiert wird
<param>	tCondition	user confirm	Gibt an, das eine Bestätigung vom Nutzer erforderlich ist (tMaker="GUI")
		task condition	Allgemeine Task Bedingung (tMaker="core", "network", "process")
	desc	[String]	Optionaler String, welcher an die entschiedene Entität weitergegeben wird und zusätzliche Informationen enthalten kann
<option>	tOption	yes	Spezifiziert den allgemeinen positiv Fall (yes, okay usw.)
		no	Spezifiziert den allgemeinen neutral Fall (no, cancel usw.)
		error	Spezifiziert den allgemeinen Fehler-Fall (error, abort usw.)
	tAction	continue	Führt die Abarbeitung des

			Skriptes normal fort
		abort	Bricht die Abarbeitung des Skriptes ab
	extCmd		Durch Komma getrennte zusätzliche Kommandos, die bei Wahl der jeweiligen Option ausgeführt werden sollen.
		informUser	Der Nutzer soll informiert werden. (verfügbar für tMaker="core", "network")
<conditions>			Sektion zum spezifizieren von Bedingungen
	bAll	[0 .. 1]	bestimmt ob alle Bedingungen erfüllt sein müssen
<conditions/condition>	tCondition	CalibNotExist	Testet ob die spezifizierte Kalibrierung nicht existiert.
		AnalogWay_Calibration_Start	Verfügbar für tMaker="network", testet ob ein AnalogWay Server erfolgreich in den VIOSO Kalibrierungsmodus umgeschaltet hat. Entweder wird der Response der letzten "send" Aktion verwendet oder eine unmittelbare "send" Aktion durchgeführt. Bei tSource="send_command" verwendet kann der Kommandostring weggelassen werden.
		AnalogWay_Calibration_Active	Verfügbar für tMaker="network", testet ob ein AnalogWay Server im VIOSO Kalibrierungsmodus ist. Entweder wird der Response der letzten "send" Aktion verwendet oder eine inplace "send" Aktion durchgeführt. Bei tSource="send_command" verwendet kann der Kommandostring weggelassen werden.
		AnalogWay_Calibration_Stop	Verfügbar für tMaker="network", testet ob ein AnalogWay Server erfolgreich den VIOSO Kalibrierungsmodus beendet

			hat. Entweder wird der Response der letzten “send“ Aktion verwendet oder eine unmittelbare “send“ Aktion durchgeführt. Bei tSource=“send_command“ verwendet kann der Kommandostring weggelassen werden.
		AnalogWay_Calibration_Transfer	Verfügbar für tMaker=“network“, sendet eine Datei zu einem AnalogWay Server. Entweder wird der Response der letzten “send“ Aktion verwendet oder eine unmittelbare “send“ Aktion durchgeführt. Bei tSource=“send_file“ verwendet kann der Kommandostring weggelassen werden.
		HTTP_Request	Evaluiert, ob ein HTTP Request erfolgreich war
		Behaviour_Finished	Evaluiert, wie ein Verhalten beendet wurde (keine normale Bedingung, wird für Wart-Aktion auf das beenden eines Verhaltens verwendet)
		Process_Finished	Evaluiert, wie ein (externer) Prozess beendet wurde
	tSource		Bestimmt die Quelle der Bedingung. Wenn nicht angegeben, wird “current“ angenommen.
		current	Der aktuelle Zustand wird verwendet.
		send_command	Das Ergebnis eines TCP Kommandos wird verwendet. Es werden die nötigen Angaben zum senden des Kommandos erwartet. Siehe Abschnitt 3.31
		send_file	Das Ergebnis einer Dateiübertragung per TCP wird verwendet. Es werden die nötigen Angaben zum senden des Kommandos erwartet. Siehe Abschnitt 3.32

	tDevice	sd	einzelnes Display
		dc	Display Compound
		sc	Super Compound
	name	[Bezeichner]	Name des Displays/ Display Compounds/ Prozess(task)
	bStdErrorCode	[0,1]	Bestimmt, wie ein möglicher Fehler-Code interpretiert wird. Wenn aktiviert, wird SPeASY Fehler-Code angenommen.

```

<define name="ConditionData" type="condition" >
  <maker tMaker="GUI" />
  <param tCondition="user confirm" desc="blabla" />
  <option tOption="yes" tAction="continue" />
  <option tOption="no" tAction="abort" />
</define>

```

Dieser Define-Block spezifiziert, daß der Nutzer über die GUI entscheiden muß, ob das aktuell ausgeführte Skript weiter fortgeführt oder abgebrochen werden soll.

```

<define name="ConditionData" type="condition" >
  <maker tMaker="core" />
  <param tCondition="task condition" desc="" />
  <option tOption="yes" tAction="abort" extCmd="informUser" />
  <option tOption="no" tAction="continue" />
  <option tOption="error" tAction="abort" extCmd="informUser" />
  <conditions bAll="0" >
    <condition tCondition="calibnotexist" tDevice="dc" name="screen0" />
  </conditions>
</define>

```

Dieser Define-Block spezifiziert, daß die Abarbeitung des Skriptes nur fortgesetzt wird, wenn das Display-Compound mit Namen „screen0“ existiert. Wenn die spezifizierte Kalibrierung nicht existiert oder ein Fehler passierte, soll der Nutzer darüber informiert werden, daß das Skript nicht weiter ausgeführt wird.

```

<define name="ConditionData" type="condition" >
  <maker tMaker="network" />
  <param tCondition="task condition" desc="" />
  <option tOption="yes" tAction="continue" />
  <option tOption="no" tAction="abort" extCmd="informUser" />
  <option tOption="error" tAction="abort" extCmd="informUser" />
  <conditions bAll="0" >
    <condition tCondition="analogway_calibration_start"
      tSource="send_command">
      <target core="192.168.2.20" iPort="80" />
    </condition>
  </conditions>
</define>

```

Dieser Define-Block spezifiziert, daß die Abarbeitung des Skriptes nur fortgesetzt wird, wenn der spezifizierte AnalogWay Server in den VIOSO Kalibrierungsmodus geschaltet wurde. Das dafür nötige Netzwerkkommando wird direkt beim evaluieren der Bedingung gesendet. Konnte der AnalogWay Server nicht in den VIOSO Kalibrierungsmodus geschaltet werden, soll der Nutzer darüber informiert werden.

3.35 *action="Wait"*

Typ des separaten Parameterblocks: **Condition**

Es können mehrere **<option>** Sektionen angegebenen werden. Siehe hierzu Abschnitt 3.34.

Derzeit verfügbar für **state="finished"**.

```

<task action="wait" name="Export0" state="finished" use="Export0Finished" />
<define name="Export0Finished" type="condition" >
  <option tOption="error" tAction="abort" extCmd="informUser" />
</define>
oder
<task action="wait" name="Export0" state="finished" >
  <option tOption="error" tAction="abort" extCmd="informUser" />
</task>

```

Dieser Define-Block spezifiziert, daß die Skript Abarbeitung abgebrochen werden und der Nutzer darüber informiert werden soll, wenn das spezifizierte Verhalten (Export0) mit einem Fehler beendet wurde.

4 zusätzliche Konfigurationsdateien

4.1 Selektor-Parameter Liste

Mit einer Selektor-Parameter Liste, kann eine Liste von Parametern definiert werden, welche über einen Selektor ausgewählt/ abgerufen werden können. Ein Parameter wird durch eine **<element>** Sektion, zusammengefaßt durch eine übergeordnete **<list>** Sektion, definiert.

Sektion	Attribut	Wert	Beschreibung
<element>	selector	[Wert]	definiert den Selektor Wert der Typ des Selektors hängt vom Attribut seltype ab
	seltype	string	der Selektor ist ein String
		integer	der Selektor definiert eine ganze Zahl (Integer)
		float	der Selektor definiert eine Fließkomma-Zahl (float)
		double	der Selektor definiert eine Fließkomma-Zahl (double)
	value	[Wert]	definiert den Paramter Wert der Typ des Paramters hängt vom Attribut valtype ab
	valtype	string	der Parameter ist ein String
		integer	der Parameter definiert eine ganze Zahl (Integer)
		float	der Parameter definiert eine Fließkomma-Zahl (float)
		double	der Parameter definiert eine Fließkomma-Zahl (double)

Beispiel:

```
<?xml version="1.0"?>
<VIOSO>
  <list>
    <element selector="2.5" seltype="float" value="c1.sps" valtype="string" />
    <element selector="3" seltype="float" value="c2.sps" valtype="string" />
  </list>
</VIOSO>
```

Die Selektor-Parameter Liste enthält zwei Einträge. Jeder Parameter definiert einen String (hier Dateiname) und kann durch eine Fließkomma-Zahl ausgewählt werden.