

Einbinden externer Systemmanagement-Programme in die CCMS-Monitoring-Architektur



SAP Monitoring Shared Library

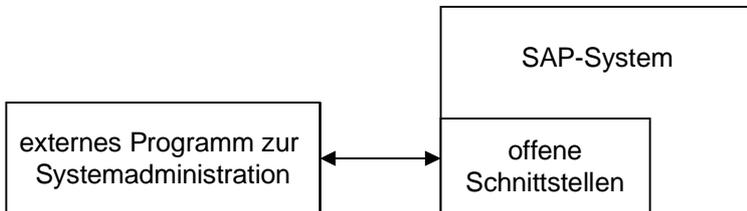
Version 1.0

Inhalt

1 Die Aufgabe externer Schnittstellen	3
2 Kurze Einführung in die CCMS-Monitoring-Architektur	4
2.1 Data Supplier	5
2.2 Alerts	5
2.3 Monitorobjekte und -attribute	6
2.6 Weiterführende Informationen zum Alert-Monitor	6
3 Konzept der externen Schnittstellen	7
3.1 Remote Function Call (RFC)	7
4 SAP Monitoring Shared Library	8
4.1 Schnittstellen	9
4.1.1 Einfache Schnittstelle	9
4.1.2 Erweiterte Schnittstelle	10
4.1.3 RFC-Schnittstelle	12
4.1.4 Funktionen	13
4.1.9 Beispielprogramm	14
5 Nagios	15
6 Beispiele der Konfigurationsdateien	18
6.1 /etc/sapmon/agent.cfg	18
6.2 /etc/sapmon/login.cfg	19
6.3 /etc/sapmon/	19
6.4 /etc/sapmon/ssh_config	19
7 Glossar	20
8 Weiterführende Informationen	22

1 Die Aufgabe externer Schnittstellen

Externe Schnittstellen sollen es ermöglichen, SAP-Installationen (insbesondere umfangreiche IT-Landschaften) durch Programme anderer Hersteller zu überwachen und zu administrieren.

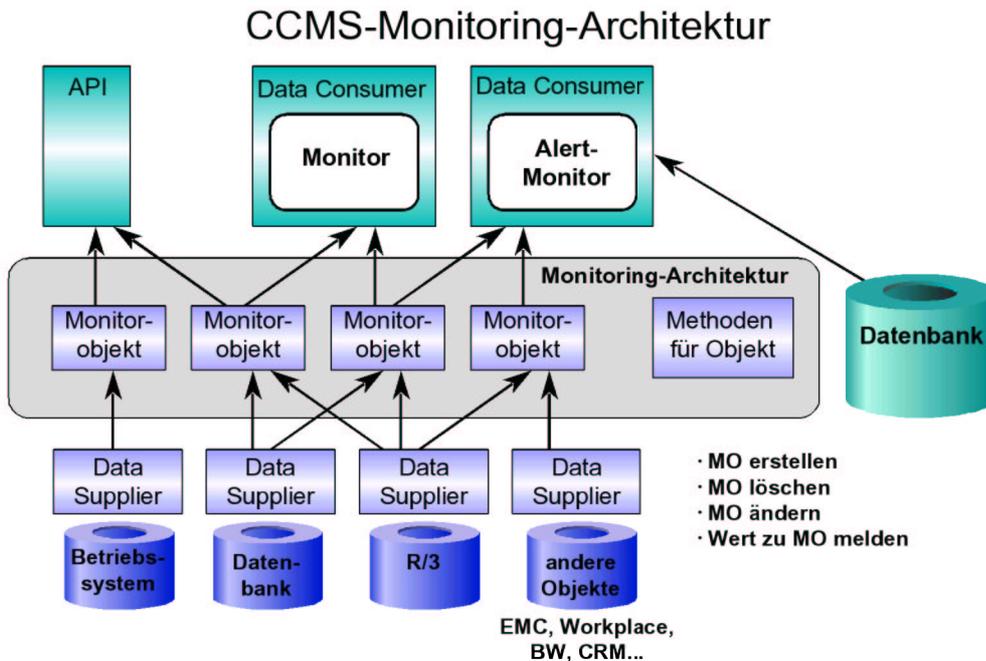


Über externe Schnittstellen können Sie SAP-Systeme einfach und nahtlos sowohl in lokale Administrationsprogramme als auch in firmenweite Systemmanagement-Infrastrukturen integrieren. Dabei kann und soll diese Integration nicht die Verwendung von CCMS-Funktionen vollständig ersetzen. Komplexe und sicherheitskritische Aufgaben zur Systemkontrolle müssen immer über CCMS durchgeführt werden. Außerdem werden Basisfunktionen von CCMS von SAP-Anwendungen und vom SAP Solution Manager benötigt und erwartet. So verwenden SAP-Anwendungen interne Funktionen zur Jobverwaltung auch dann, wenn Sie ein externes Programm zur Jobverwaltung und -einplanung verwenden.

Das Ziel der Integration ist es, dem Anwender eine homogene Infrastruktur zur Anzeige der nötigen Informationen zu ermöglichen. Das Ziel der Schnittstellen ist es, den Informationsfluss zwischen SAP-Systemen und externen Programmen zu erleichtern. So können externe Programme zusätzliche Flexibilität bei der Ergänzung grundlegender CCMS-Funktionen bieten.

2 Kurze Einführung in die CCMS-Monitoring-Architektur

Im Alert-Monitor werden Problemmeldungen (sog. Alerts) in einer Baumstruktur angezeigt und einer Gewichtung sowie einer Farbe (Gelb für eine Warnung, Rot für ein Problem) zugeordnet. Sie können hier den aktuellen Status Ihres Systems erkennen und Alerts bearbeiten. Der Alert-Monitor basiert auf der mit SAP-Basis-Release 4.0 eingeführten Monitoring-Architektur:



Die CCMS-Monitoring-Architektur ist kein monolithisches Überwachungs- und Verwaltungsprogramm. Vielmehr bietet sie einen flexiblen Rahmen, dem problemlos weitreichende Überwachungs- und Verwaltungsfunktionen hinzugefügt werden können. Die Bestandteile der Monitoring-Architektur arbeiten weitgehend unabhängig voneinander und können vor allem auch unabhängig voneinander weiterentwickelt und angepasst werden:

- **Data Supplier**

Ein Datasupplier ist ein Programm, das der Monitoring-Architektur Daten zu überwachten Objekten liefert. Diese bietet eine Infrastruktur zum Erfassen und Verwalten von Systeminformationen an. Die Monitoring-Architektur vergleicht die Messwerte der Data Suppliers ständig mit Schwellenwerten und gibt beim Über- oder Unterschreiten einen Alert aus.

- **Data Consumer**

Ein Data Consumer ist ein Programm, das Daten aus der Monitoring-Architektur liest; es zeigt die von den Data Suppliers an die Monitoring-Architektur übergebenen Informationen an. SAP liefert sowohl mit dem Alert-Monitor den standardmäßigen Data Consumer als auch andere speziellere Monitore, die die Daten der Monitoring-Architektur verwenden.

- **Monitorobjekte und -attribute**

Ein Monitorobjekt bezeichnet ein Objekt, das überwacht werden soll; ein Monitorattribut bezeichnet einen Informationstyp, der zu einem Monitorobjekt gemeldet werden soll (siehe Monitorobjekte und -attribute).

Der Alert-Monitor bietet außerdem die für die Überwachung des Systems notwendigen Verwaltungsmethoden. So können Sie die Schwellenwerte für Alerts einstellen sowie Autoreaktions- und Analysemethoden hinzufügen oder anpassen: Wenn ein Alert ausgelöst wird, reagieren Autoreaktionsmethoden automatisch; mit einer Analysemethode können Sie die Ursache eines Alerts untersuchen, ohne den Alert-Monitor zu verlassen. Außerdem enthält die Monitoring-Architektur Werkzeuge zur Verwaltung und Archivierung der Alerts.

2.1 Data Supplier

Data Suppliers liefern alle Werte, die im Alert-Monitor angezeigt werden. Sie gehören jeweils zu den einzelnen Systemkomponenten und legen Monitorobjekte und -attribute an, die der Monitoring-Architektur Werte melden. Diese Werte werden in Monitorsammlungen angezeigt. Die Monitoring-Architektur wird mit den Data Suppliers für die wichtigsten Komponenten Ihres SAP-Systems und seiner Umgebung ausgeliefert und ist damit nach der Konfiguration direkt einsatzbereit.

2.2 Alerts

Der Alert-Monitor generiert in den überwachten Systemen anhand von Schwellenwerten und Regeln Alerts, sobald deren Zustand von der Norm abweicht. Die Alerts lenken Ihre Aufmerksamkeit auf kritische Situationen, so dass Sie nicht selbst danach suchen müssen. Autoreaktionsmethoden informieren Sie über ein Problem oder reagieren anderweitig darauf, auch wenn Sie nicht im Alert-Monitor arbeiten.

Für den Systemverwalter ist ein Alert eine Störmeldung oder Serviceanforderung. Reagieren Sie auf jeden Alert. Untersuchen Sie den Alert mit Hilfe der Analysemethoden des Alert-Monitors. Wenn Sie das Problem gelöst haben oder sicher sind, dass Sie es ohne Bedenken ignorieren können, setzen Sie den Alert auf *erledigt*. Der Alert wird aus der Anzeige gelöscht und in der Alert-Datenbank gesichert.

Ein Alert ist eindeutig einem Monitorbaumelement (MTE) im Monitorbaum zugeordnet. Der Alert wird erstellt, wenn Werte von einem Data Supplier an ein MTE gesendet werden. Der Alert-Monitor bewertet die eingehenden Informationen im Vergleich mit den Alert-Schwellenwerten. Wenn die eingehenden Daten einen Schwellenwert über- oder unterschreiten, löst der Alert-Monitor einen Alert aus. Der Alert-Monitor erzeugt für jeden von der Norm abweichenden Zustand einen Alert und verfolgt diesen, bis er auf den Status *erledigt* gesetzt wird.

Alert-Farben und ihre Bedeutung

Farbe	Bedeutung
rot	Problem oder Fehler
gelb	Warnung
grün	alles in Ordnung
grau	keine Informationen verfügbar

2.3 Monitorobjekte und -attribute

Ein Monitorobjekt repräsentiert eine Komponente der IT-Umgebung, die überwacht werden soll, z. B. die CPU eines Servers, das Dialogsystem oder die Hintergrundverarbeitung. Monitorattribute sind Werte, Status oder Texte, die zu diesem Objekt gemeldet werden, etwa die CPU-Auslastung oder die Antwortzeit im Dialogsystem. Einem Monitorattribut kann ein Alert zugewiesen sein. Die Auswahl der Monitorobjekte erfolgt durch Data Supplier, die zu allen Bereichen des Systemmanagements existieren.

Monitorobjekte und Attribute werden im Alert-Monitorbaum als einzelne Knoten in einem hierarchischen Baum angezeigt. Wenn die der Monitoring-Architektur gemeldeten Daten die definierten Alert-Schwellenwerte über- oder unterschreiten, wird in dem entsprechenden MTE ein Alert ausgelöst.

Typen von Monitorattributen

Attributtyp	Bedeutung
 Performance-Attribut	sammelt gemeldete Performance-Werte und errechnet das Mittel
 Statusattribut	meldet Fehlermeldungstexte und Alert-Status
 Protokollattribut	prüft Protokoll- und Trace-Dateien (diese Attribute können einen vorhandenen Protokollmechanismus verwenden, z. B. das SAP-Systemprotokoll, oder sie können von einer Anwendung für die Implementierung eines eigenen Protokolls verwendet werden)
 Textattribut	ermöglicht es einem Data Supplier, Informationen ohne Alertbewertung zu melden; der Text kann nach Bedarf aktualisiert werden

2.4

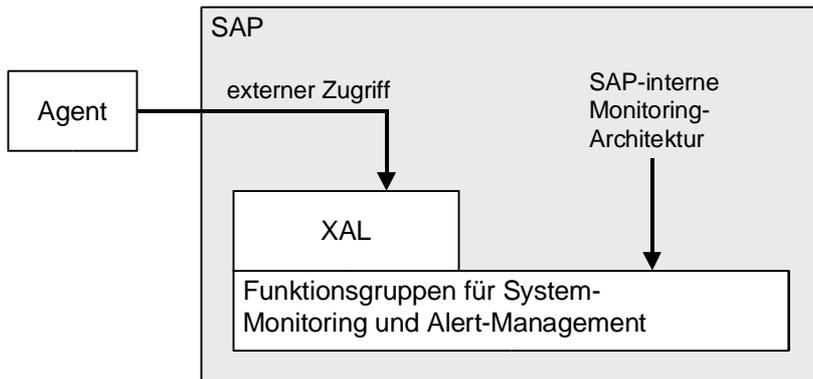
2.5

2.6 Weiterführende Informationen zum Alert-Monitor

- Eine ausführliche Einführung in die Konzepte und die Verwendung der Monitoring-Architektur, insbesondere die Bedienung des Alert-Monitors, finden Sie unter **[AIMon]**.
- Eine ausführliche Beschreibung der von SAP ausgelieferten Standardmonitore mit ihren Monitorbaumelementen finden Sie unter **[StdMon]**

3 Konzept der externen Schnittstellen

Die verwendete XAL-Schnittstelle ist Teil eines Schnittstellenpakets für externe Systemmanagement-Programme. SAP-seitig bestehen diese Schnittstellen aus einer Sammlung von Funktionsbausteinen. Externe Programme können diese Bausteine über RFC (Remote Function Call). Alle RFC-Schnittstellen verwenden dieselben Funktionsgruppen wie die entsprechenden internen SAP-Funktionen. Dies gilt natürlich nicht für die XMI-Schnittstelle, da für diesen Fall keine Funktionen zur internen Nutzung existieren.



3.0.1

3.1 Remote Function Call (RFC)

Funktionen innerhalb und außerhalb von SAP-Systemen verwenden RFC, um miteinander zu kommunizieren. Technisch gesehen ist die RFC-Funktionalität als Programmbibliothek verfügbar (geschrieben in der Programmiersprache C), die auf allen von SAP unterstützten Betriebssystem-Plattformen zur Verfügung steht.

Funktionsweise:

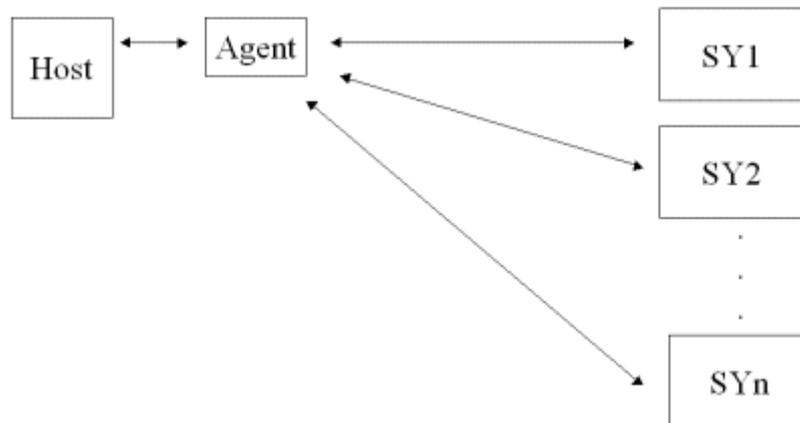
Bei externen Systemmanagement-Programmen ist das Programm gegenüber einem SAP-System ein Klient, dem ein Anwendungsserver ein Service zur Verfügung stellt. Dieser Service umfasst die Lieferung von internen Systeminformationen sowie die Ausführung einzelner Systemmanagement-Funktionen.

RFC arbeitet in einzelnen Sessions – der Anwender eröffnet eine Session, führt die entsprechenden Aufgaben durch und schließt die Session wieder. Die Eröffnung erfordert dabei eine reguläre Anmeldung an dem SAP-System mit Benutzer und Kennwort, mit dem er vom SAP-System erkannt wird. Dieser Benutzer muss also innerhalb des SAP-Systems existieren und für die gewünschten Aktivitäten der RFC-Session über die notwendigen Berechtigungen verfügen.

4SAP Monitoring Shared Library

Die SAP Monitoring Shared Library ist ein Wrapper um die XAL-Schnittstelle und RFC-Verbindung. Sie erlaubt einen einfachen Zugriff auf die Daten im Monitoring Segment.

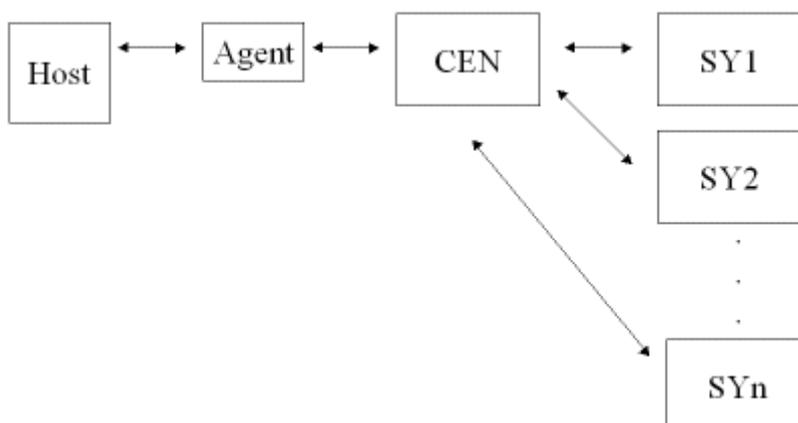
Mithilfe der Shared Library kann man die Daten ganzen Systemlandschaften auslesen, wobei es verschiedene Ansätze gibt:



Es gilt eine Systemlandschaft mit n Systemen SY1 .. SYn zu überwachen.

Der Agent dient als Schnittstelle zwischen dem überwachenden Host und den n Systemen.

Bei diesem Ansatz sind alle Systeme gleichberechtigt, der Agent muss sich in jedes System einloggen, die Daten sammeln und wieder ausloggen. Für jedes System müssen Anmeldedaten für den Agenten hinterlegt sein.



Ein zweiter Ansatz geht über ein zentrales System. Der Agent kommuniziert nur mit dem zentralen System, das die anderen Systeme überwacht. Der Agent loggt sich in das zentrale System ein und sammelt die Daten von allen Systemen. Das zentrale System muss in SAP konfiguriert sein.

4.1 Schnittstellen

Die Shared Library bietet zwei verschiedene Schnittstellen:

- Einfache Schnittstelle ohne direkten CCMS-Monitoring Bezug (eingeschränkte Funktionalität)
- Erweiterte Schnittstelle (entspricht CCMS-Monitoring)

4.1.1 Einfache Schnittstelle

Welche Daten von den überwachten Systemen gesammelt werden, wird in der Datei /etc/sapmoni/agent.cfg festgelegt. Dort wird in Templates festgelegt, welche Daten zu sammeln sind. In der agent.cfg befinden sich ein oder mehrere Templates folgendem Aufbau:

```
[TEMPLATE_<NAME>]

DESCRIPTION = Beschreibung des Templates           z.B. Einfaches Template

SYSTEM = <SID >                                   z.B. BCE | BC? | B*

APPL-SERVER = <APPL-SERVER>                       z.B. bcemain_BCE_26 | bce* | *

VALUE = <VALUE>                                   z.B. DIALOG_RESPONSE_TIME
```

z.B.

```
[TEMPLATE_TEST]

SYSTEM = BCE

APPL-SERVER = *

VALUE = DIALOG_RESPONSE_TIME
```

Dieses Template names TEST würde die Dialog Response Zeit von allen Applikationserver des Systems BCE zurückliefern.

Welche Werte als Value angegeben werden können steht in der Datei /etc/moni_tr.cfg.

4.1.2 Erweiterte Schnittstelle

Die Konfiguration erfolgt auch hier über die Datei /etc/sapmoni/agent.cfg.

Jedoch ist der Aufbau der Templates anders:

```
[TEMPLATE_<NAME>]

DESCRIPTION = Beschreibung des Templates           z.B. Einfaches Template

MONI_SET_NAME= <Monitorsammlung>                  z.B. "SAP ADMIN TEMPLATE"

MONI_NAME= <Monitor der Sammlung>                 z.B. "Operating System"

MAX_TREE_DEPTH = n                               z.B. 3

PATTERN_0=SID\Kontext\Monitorobjekt\Monitorattribut

           z.B. BCE\bcemain_BCE_26\CPU\CPU_UTILIZATION
```

Auch hier wird dem Template ein Name gegeben.

MONI_SET_NAME gibt die Monitorsammlung an, in der der Monitor MONI_NAME geladen wird. Der optionale Parameter MAX_TREE_DEPTH gibt an, bis in welche Tiefe der Monitor geladen wird (default 0 = ganzer Monitor).

Hiermit wird der komplette Monitor übertragen. Da man aber nur an bestimmten Werten interessiert ist, kann man diese mit den PATTERN_X einschränken. Die Pattern sind ein Viertupel bestehend aus

SID\Kontext\Monitorobjekt\Monitorattribut

Alle Knoten, die aus dem Monitor geladen werden, werden mit den Pattern verglichen. Erfüllt der Knoten den Vergleich nicht, wird er aus der Liste gelöscht.

In unserem Beispiel wird der Knoten CPU_UTILIZATION auf dem Appl.Server bcemain vom System BCE (PATTERN_0) geladen.

```
PATTERN_0="BC?\*\*\DialogResponseTime"
```

Hier werden alle Dialogantwortzeiten aller Appl.Server der Systeme BC? geladen.

```
[TEMPLATE_0]

DESCRIPTION = Einfaches Template

MONI_SET_NAME="SAP ADMIN TEMPLATE"

MONI_NAME="Operating System"

PATTERN_0="BCE\*"
```

Hier werden alle Knoten des Systems BCE geladen, die sich im angegebenen Monitor befinden.

Dadurch, dass zuerst die Monitore geladen und die Knoten übertragen werden, entsteht ein Overhead.

Deswegen sollte man sich im Alertmonitor (TX RZ20) seinen eigenen Monitor mit allen wichtigen Knoten erstellen und diesen ganz übertragen (PATTERN weglassen -> PATTERN = *).

Dadurch entsteht kein Overhead und man kann alle wichtigen Daten auf einmal übertragen.

Die einfache Schnittstelle übersetzt lediglich ihren Syntax in die erweiterte Form. Deshalb bildet sie nur eine Teilmenge ab.

Aus

```
[TEMPLATE_TEST]

SYSTEM = BCE

APPL-SERVER = *

VALUE = DIALOG_RESPONSE_TIME
```

wird

```
[TEMPLATE_TEST]

MONI_SET_NAME="SAP ADMIN TEMPLATE"

MONI_NAME="Dialog Overview"

PATTERN_0="BCE\*\DIALOG\DIALOGRESPONSETIME"
```

In der Datei /etc/sapmoni/moni_tr.cfg steht, wie die Substitution stattfinden soll:

```
DIALOG_RESPONSE_TIME#SAP ADMIN TEMPLATE# Dialog Overview#DIALOG\DIALOGRESPONSETIME
```

Das bedeutet, dass für VALUE=DIALOGRESPONSETIME der Monitor Dialog Overview aus der Monitorsammlung SAP ADMIN TEMPLATE gestartet werden soll.

Das Pattern wird aus SYSTEMAPPL-Server (Kontext)\Rest (hier DIALOG\DIALOGRESPONSETIME) zusammengebaut.

Durch Einfügen eigener Zeilen in der Datei /etc/sapmoni/moni_tr.cfg, kann die einfache Schnittstelle erweitert werden.

```
int cfg_init(AlMoniHandle_T **MoniHandle, int * NrOfMoniTempl);
```

Liest die agent.cfg ein und gibt ein Array von MoniHandles zurück. Die MoniHandles repräsentieren einen Monitor mit n Knoten.

4.1.3 RFC-Schnittstelle

Die Verbindung zum R/3-System erfolgt per RFC.
Hier müssen nur die System- und Logon-Daten hinterlegt werden.

Die Logon-Daten werden in der Datei /etc/sapmon/login.cfg hinterlegt. Auch hier werden Templates verwendet.

Am Anfang muss man eine Verbindung aufgebaut werden. Dies funktioniert über die Funktion

```
Int Connection_Init ( char *Login_Template, RFC_HANDLE * handle);
```

Als Argumente bekommt die Funktion *Login_Template und liefert einen RFC_HANDLE zurück.

Die login.cfg hat folgenden Aufbau:

```
[LOGIN_<templatename>]
LOGIN=<Argumente>
```

Argumente:

```
-D -u -p -c |
-d -u -p -c -h -s [ -l -t -g -x ]

-d <destination>
-D <destination, uses saprfc.ini >
-u <userid>
-p <password>
-c <client>
-l <language>
-t <Trace>
-h <hostname>
-s <system number>
-g <gateway host>
-x <gateway service>
```

Hier hat man die Entscheidung, ob man die saprfc.ini nutzen will oder nicht.
In der saprfc.ini sind die Verbindungsdaten abgelegt, man muss nur die Logindaten mitgeben.

Will man die saprfc.ini benutzen, lauten die Argumente: -D <Destination> -u <user> -p <passwd> -c <client>.
Die Destination muss in der saprfc.ini existieren.

Ohne saprfc.ini muss man alle Eingaben machen:

```
-d <Destination> -u <user> -p <passwd> -c <client> -h <hostname> -s <system number>
```

4.1.4 Funktionen

4.1.5

Hat man über `cfg_init(...)` die `agent.ini` eingelesen und die RFC-Verbindung über `Connection_Init (...)` aufgebaut, kann man das R/3 System auslesen.

Dazu gibt es zwei Methoden:

```
int generate_monitor(RFC_HANDLE * handle, AlMoniHandle_T *MoniHandle, int *
NrOfMoniTempl);
int refresh_monitor (RFC_HANDLE * handle, AlMoniHandle_T *MoniHandle, int *
NrOfMoniTempl);
```

Mit `generate_monitor` werden die Knoten aus dem R/3 geladen und die Werte gefüllt. Über `refresh_monitor` können geladenen Monitore aktualisiert werden, d.h. die Werte werden aktualisiert.

Man gibt den Funktionen einen `RFC_HANDLE`, ein Array von `MoniHandles` und die Anzahl der `MoniHandles` mit und bekommt die Strukturen der `MoniHandles` gefüllt.

Mit `get_tree_information(AlMoniHandle_T MoniHandle, AlMoniNode_T * nodes, int * NrOfNodes);` bekommt man ein Array von Knoten zurück, die zu dem `MoniHandle` (= Monitor) gehören.

Weiterhin gibt es noch

```
int GetName (AlMoniHandle_T MoniHandle);
int getDescripton (AlMoniHandle_T MoniHandle);
```

Die einem den Namen und den Beschreibungstext des Templates angeben.

Zum Schluss muss man die RFC-Verbindung trennen, dies geschieht über

```
XMI_Logoff(RFC_HANDLE * handle);
```

4.1.6

Auch gibt es Funktionen zum Starten und Stoppen eines R/3s:

```
int start_sap (char * SID, char * host, char * systemnummer);
int stop_sap (char * SID, char * host, char * systemnummer);
```

4.1.7

4.1.8

4.1.9 Beispielprogramm

4.1.10

```

int main(int argc, char **argv) {
    int i,j;

    RFC_HANDLE rfc_handle;
    AlMoniHandle_T *MoniHandle;
    int   NrOfMoniTempl;

    AlMoniNode_T *nodes;
    unsigned int NrOfNodes;

    if (argc < 3) {
        usage(argv[0]);
        return AG_INVALID_ARGUMENTS;
    }

    if ( cfg_init(&MoniHandle, &NrOfMoniTempl) >= AG_FIRST_ERROR) {
        printf("Could not initialize INI-File\n");
        return AG_CALL_INVALID;
    }

    if( Connection_Init (argc, argv, &rfc_handle) >= AG_FIRST_ERROR) {
        printf("Could not log on\n");
        return AG_CALL_INVALID;
    }

    for (i = 0; i < NrOfMoniTempl; i++) {
        generate_monitor(&rfc_handle, &(MoniHandle[i]), 1 );
        get_tree_information(MoniHandle[i], &nodes, &NrOfNodes);

        for (j = 0; j < NrOfNodes; j++) {
            printf("MO: %s, MA: %s, Value %s\n",
                nodes[j].Objectname, nodes[j].Mtname, nodes[j].currentValue);
        }
    }

    while(1) {
        for (i = 0; i < NrOfMoniTempl; i++) {
            refresh_monitor( &rfc_handle, MoniHandle, NrOfMoniTempl );
            get_tree_information(MoniHandle[i], &nodes, &NrOfNodes);

            for (j = 0; j < NrOfNodes; j++) {
                printf("MO: %s, MA: %s, Value %s\n",
                    nodes[j].Objectname, nodes[j].Mtname, nodes[j].currentValue);
            }
        }
        sleep(10);
    }

    //Log off
    XMI_Logoff(&rfc_handle);

    return SAP_OK;
}

```

Das Programm liest die Daten ein und refreshed alle 10s die Daten.

5 Nagios

Eine Nutzung der Shared-Lib ist die Nagios-Implementierung.

Zuerst muss Nagios installiert werden (hierfür die SAP-Version verwenden, da Nagios nicht die volle Funktionalität bietet).

Mithilfe der Library wurden Datensammler (Plugins) geschrieben, die Daten aus dem R/3 ziehen.

Die Datensammler sind:

check_sap <Monitemplate> <RFC-Template>	Liefert die Daten aus dem Monitemplate.
check_sap_multiple <Monitemplate> <RFC-Template>	Liefert die Daten aus dem Monitemplate. Kann mehrere Werte auf einmal anzeigen
check_sap_multiple_no_thr <Monitemplate> <RFC-Template>	Wie check_sap_multiple, es sind aber keine Schwellwertänderungen möglich. Das erlaubt mehr Werte anzuzeigen, da kein Platz für die Links vonnöten ist.
check_sap_system <SID> <RFC-Template>	Liefert die App.Server und Status der App.server. CCMS-Ping muss installiert sein.
check_sap_instance <Appl.Server><RFC-Template>	Liefert Standard-Daten zum Appl.Server. CCMS-Ping muss installiert sein

Die Implementierung läuft Nagios-konform:

- Plugins in das Plugin-Verzeichnis kopieren
- Zu überwachende Rechner in die hosts-Datei eintragen:

```
define host{
    host_name bcemain
    alias    bcemain_BCE_26
    address 10.17.72.234
    max_check_attempts 5
    notification_interval 30
    notification_period 24x7
    notification_options d,u,r
}

```

Wichtig ist, dass der Alias SAP-konform, <Appl.server>_<SID>_<Systemnr>, ist.

- Methoden/Plugins eintragen (checkcommands.cfg)

```
# 'SAP Monitoring' command definition
define command{
    command_name    check_sap
    command_line    $USER1$/check_sap $ARG1$ $ARG2$
}
# 'SAP Monitoring' command definition
define command{
    command_name    check_sap_multiple
    command_line    $USER1$/check_sap_multiple $ARG1$ $ARG2$
}
# 'SAP Monitoring' command definition
define command{
    command_name    check_sap_system
    command_line    $USER1$/check_sap_system $ARG1$ $ARG2$
}
# 'SAP Monitoring' command definition
define command{
    command_name    check_sap_instance
    command_line    $USER1$/check_sap_instance $ARG1$ $ARG2$
}
```

- Methoden den Hosts zuordnen (services.cfg)

```
define service{
    use                generic-service
    host_name          BCE
    service_description System BCE
    is_volatile        0
    check_period       24x7
    max_check_attempts 3
    normal_check_interval 300
    retry_check_interval 1
    contact_groups     linux-admins
    notification_interval 300
    notification_period 24x7
    notification_options w,u,c,r
    check_command      check_sap!bce!BCE!
}
define service{
    use                generic-service
    host_name          PWDF0487
    service_description Standard for App.server
    is_volatile        0
    check_period       24x7
    max_check_attempts 3
    normal_check_interval 300
    retry_check_interval 1
    contact_groups     linux-admins
    notification_interval 300
    notification_period 24x7
    notification_options w,u,c,r
}
```

```

check_command
}

```

```
check_sap_instance!PWDF0631_BCE_26!BCE!
```

Für Details siehe Nagios Dokumentation.

Weiterhin bietet die Nagios-Implementierung die Möglichkeit das SAP-System zu Starten/Stoppen, eine SAP-GUI und ssh-Zugang zu dem System zu öffnen und Schwellwerte zu ändern.

Host	Service	Status	Last Check	Next Check	Performance	Output
is3532	Standard for App.server	UNKNOWN	07-30-2002 15:46:24	0d 1h 22m 0s	3/3	Dialog Response Time 0 msec Users Logged On 0 Users
is3532	bcemain	OK	07-30-2002 15:46:24	0d 0h 1m 38s	1/3	BCE bcemain_BCE_26 Dialog ResponseTime 698 msec
is3532	BCE Dialog	OK	07-30-2002 15:46:03	0d 0h 3m 18s	1/3	BCE bcemain_BCE_26 Dialog ResponseTime 698 msec
is3532	bcemain Network Time	CRITICAL	07-30-2002 15:45:55	0d 3h 12m 0s	3/3	BCE PWDf0631_BCE_26 Dialog FrontEndNetTime 6701 msec BCE PWDf0632_BCE_26 Dialog FrontEndNetTime 5317 msec BCE bcemain_BCE_26 Dialog FrontEndNetTime 8 msec BCE hw1112_BCE_26 Dialog FrontEndNetTime 0 msec
is3532	bcemain Network Time Multiple	CRITICAL	07-30-2002 15:45:43	0d 3h 11m 50s	3/3	BCE PWDf0631_BCE_26 Dialog FrontEndNetTime 6701 msec BCE PWDf0632_BCE_26 Dialog FrontEndNetTime 5317 msec BCE bcemain_BCE_26 Dialog FrontEndNetTime 8 msec BCE hw1112_BCE_26 Dialog FrontEndNetTime 0 msec
is3532	dev/hda1 Free Space	WARNING	07-30-2002 15:41:29	0d 18h 36m 53s	3/3	Disk "20k" not mounted or non-existent
is3532	WAS Appl.	OK	07-30-2002 15:45:55	0d 17h 58m 3s	1/3	WAS is3532_WAS_18 Dialog ResponseTime 0 msec

Nach einem Klick auf den Hostnamen, öffnet sich ein Fenster, in dem man das SAP-System starten und stoppen kann, sowie eine SAP-GUI oder ssh öffnen kann.

Durch einen Klick auf einen Wert, öffnet sich ein Schwellwert-Pflegefenster.

Wichtig ist, dass der Alias SAP-konform gesetzt ist. Anderenfalls stehen die SAP-Funktionalitäten nicht zur Verfügung.

Es ist wichtig die Templates so anzupassen, dass nicht zu viele Werte pro Template gemeldet werden, da die Anzahl der Buchstaben pro Service begrenzt sind.

6 Beispiele der Konfigurationsdateien

6.1/etc/sapmon/agent.cfg

```
[TEMPLATE_0]

DESCRIPTION="Load Average"

MONI_SET_NAME=SAP CCMS Admin Workplace

MONI_NAME="Operating System"

MAX_TREE_DEPTH=0

PATTERN_0="BCE\bcmmain_BCE_26\CPU\5minLoadAverage"

PATTERN_0="*\*\CPU\5minLoadAverage"
```

```
[TEMPLATE_1]

MONI_SET_NAME=SAP CCMS Admin Workplace

MONI_NAME="Operating System"

MAX_TREE_DEPTH=0

PATTERN_2="BCE\bcmmain_BCE_26\CPU\CPU_U*"
```

```
[TEMPLATE_DIALOG]

VALUE=DIALOG_RESPONSE_TIME
```

```
[TEMPLATE_DIALOG1]

SYSTEM=BCE

APPL-SERVER=bcmmain*

VALUE=DIALOG_RESPONSE_TIME
```

```
[TEMPLATE_NETWORK1]

MONI_SET_NAME="SAP CCMS Monitor Templates"

MONI_NAME="Dialog Overview"

PATTERN_0="BCE\*\Dialog\FrontEndNetTime"
```

6.2/etc/sapmon/login.cfg

```
[LOGIN_BCE]
LOGIN=-d BCE -u ag_t -p *** -h bcemain -s 26

[LOGIN_WAS]
LOGIN=-d WAS -h ls3532 -s 18 -u developer -p ****
```

6.3/etc/sapmon/

```
DIALOG_RESPONSE_TIME#SAP CCMS Monitor Templates#Dialog Overview#Dialog\ResponseTime#
CHECK_SAP_SYSTEMS#SAP CCMS Monitor Templates#Availability and Performance Overview##
```

6.4/etc/sapmon/ssh_config

```
#      $OpenBSD: ssh_config,v 1.10 2001/04/03 21:19:38 todd Exp $

# This is ssh client systemwide configuration file.  See ssh(1) for more
# information.  This file provides defaults for users, and the values can
# be changed in per-user configuration files or on the command line.

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for various options

Host *
# ForwardAgent no
# ForwardX11 no
# RhostsAuthentication no
# RhostsRSAAuthentication yes
# RSAAuthentication yes
# PasswordAuthentication no
# FallBackToRsh no
# UseRsh no
# BatchMode no
# CheckHostIP yes
# StrictHostKeyChecking yes
# IdentityFile ~/.ssh/identity
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_rsa
# Port 22
# Protocol 2,1
# Cipher blowfish
# EscapeChar ~
```

Wird ein SAP-System gestartet oder gestoppt, funktioniert dies über ssh.
Man hat die Möglichkeit sich die Ausgabe durch X11-Forwarding anzuzeigen.

Hat man keine Möglichkeit für X11-Forwarding, muss eine Host-Authentifizierung gepflegt sein und die Passwort-Authentifizierung ausgeschaltet sein.
Die Konfiguration (Passwort-Authentifizierung ausschalten) erfolgt über die Datei /etc/sapmon/ssh_config.

7Glossar

Alert	Störmeldung im CCMS-Alert-Monitor und in der Monitoring-Architektur. Alerts lenken die Aufmerksamkeit des Systemadministrators auf kritische oder potenziell gefährliche Situationen in einem SAP-System. Der CCMS-Alert-Monitor wertet die von den Data Suppliers gemeldeten Werte aus und vergleicht sie mit den vordefinierten Alert-Schwellenwerten. Wenn die eingehenden Daten gegen einen Schwellenwert verstoßen, löst der Alert-Monitor einen Alert aus. Der Alert-Status jedes MTEs in einem Monitorbaum wird klar angezeigt und entsprechend farblich gekennzeichnet.
Alert-Monitor	Standard-Data-Consumer, der in einer hierarchischen Struktur die Messwerte und die Alerts der überwachten Monitorobjekte und -attribute anzeigt.
Data Consumer	Programm, das die Daten aus der Monitoring-Architektur verwendet und anzeigt (z. B. die CCMS-Monitore).
Data Supplier	Programm oder Transaktion, das Informationen über das SAP-System und seine Umgebung sammelt und diese in der CCMS-Monitoring-Architektur registriert. Es gibt folgende Arten von Data Suppliers: <ul style="list-style-type: none"> • Aktive Data Suppliers werden von einer Komponente innerhalb oder außerhalb des SAP-Systems ausgeführt, die Daten in die Monitoring-Architektur melden möchte. • Passive Data Suppliers (oder Datensammelmethoden) werden periodisch von der Monitoring-Architektur zum Sammeln von Daten ausgeführt.
Farbe	Kategorisierung von Alerts. Ein roter Alert weist auf einen Fehler oder ein Problem hin, während ein gelber Alert eine Warnung bedeutet. Der Alert-Status grün bedeutet, dass alles ordnungsgemäß läuft.
Monitor	Sammlung von MTEs, die in einem hierarchischen Baum angeordnet sind.
Monitorattribut (MA)	Grundelement innerhalb des Monitorbaums. Beschreibt die Eigenschaften der Monitorobjekte; es gibt die folgenden Arten von Monitorattributen: <ul style="list-style-type: none"> • Performance-Attribute • Statusattribute • Protokollattribute • Textattribute <p>Jedes dieser Attribute hat spezifische Eigenschaften, allerdings können bei allen Attributen (außer Textattributen) Alerts ausgelöst werden.</p>
Monitorbaum	Baumstruktur, die die von der Monitoring-Architektur überwachten Systemkomponenten zeigt. Je nach Definition kann ein Monitorbaum nur einen Teil oder alle der verfügbaren Knoten (oder MTEs) anzeigen.
Monitorbaumelement (MTE)	Knoten im Monitorbaum, von denen drei Arten existieren: <ul style="list-style-type: none"> • Monitorattribute • Monitorobjekte

	<ul style="list-style-type: none"> • Sammelknoten
Monitorkontext	Gruppe von Monitorobjekten, die unter einer Überschrift zusammen in einem Baum angeordnet sind. Jeder unabhängige Baum bildet einen Monitorkontext innerhalb eines Monitorsegments. Der Monitorbaum erhält seinen Namen von dem Wurzelement des Baums.
Monitorobjekt	MTE, das verschiedene Monitorattribute enthält, die zu demselben realen Objekt gehören (z. B. Festplatte oder Komponenten im SAP-System).
Monitorsammlung	Reihe von Monitoren, die in einem hierarchischen Baum angezeigt werden. Mit Monitorsammlungen werden die einzelnen Monitore visuell gruppiert.
Monitorsegment (MS)	„Container“ mit allen aktiven Komponenten und Daten, die in der Monitoring-Architektur benötigt werden. Das Monitorsegment wird durch zwei technische Implementierungen realisiert: <ul style="list-style-type: none"> • gemeinsamer Speicher (speicherorientierte Datei) • Reihe von Tabellen in der Datenbank
MTE-Klasse	Gruppierung von MTEs. MTE-Klassen enthalten Informationen zu allgemeinen Eigenschaften und Methodenzuordnungen für Monitorobjekte und -attribute.
Performance-Attribut	Monitorattribut, das Performance-Werte sammelt und mittelt, die der Monitoring-Architektur gemeldet wurden. Wenn diese Werte gegen die eingestellten Schwellenwerte verstoßen, wird ein Alert ausgelöst.
Protokollattribut	Monitorattribut, das wie ein Statusattribut Nachrichten mit einer Alert-Bewertung enthält. Ein Protokollattribut kann mehrere Nachrichten enthalten. Es liefert das Mittel, um Protokoll- und Trace-Dateien für die Aufnahme in der Monitoring-Architektur zu instrumentieren.
Schwellenwert	Wert, dessen Über- oder Unterschreitung in einem Performance-Attribut einen Alert auslöst.
Statusattribut	Monitorattribut, in das jeweils ein Fehlermeldungstext und Alert-Status gemeldet werden kann.
Textattribut	Monitorattribut, in das ein Data Supplier technische Informationen melden kann. Der Text kann nach Bedarf aktualisiert werden. Es können jedoch keine Werte gemeldet und keine Alerts ausgelöst werden.
virtuelles MTE	MTE zur optischen Strukturierung von Monitoren. Für virtuelle MTEs können selbst keine Werte gemeldet werden, sie sind auch nicht im Monitorsegment gespeichert.

8 Weiterführende Informationen

- [AIMon]** Online-Dokumentation Alert-Monitor
- Rufen Sie hierzu die SAP-Bibliothek auf, und wählen Sie *mySAP-Technology-Komponenten* → *SAP Web Application Server* → *Computing Center Management System (BC-CCM)* → *Monitoring im CCMS* → *Alert-Monitor*.
- [StdMon]** Voreingestellte Monitore des Alert-Monitors
- Starten sie den *SAP Service Marketplace*, und springen Sie in den Bereich *System Management and Alert Monitoring* (<http://service.sap.com/monitoring>). Wählen Sie *Help for Selected Topics*, und wählen in der Liste das Dokument *Voreingestellte Monitore*.
- [Agents]** Dokumentation zu den CCMS-Agenten
- Starten sie den *SAP Service Marketplace*, und springen Sie in den Bereich *System Management and Alert Monitoring* (<http://service.sap.com/monitoring>). Wählen Sie *Technical Information*, und wählen in der Liste das Dokument *CCMS-Agenten: Eigenschaften, Installation und Betrieb*.
- [DataSup]** Anleitung zur Erstellung eines Data Supplier
- Starten sie den *SAP Service Marketplace*, und springen Sie in den Bereich *System Management and Alert Monitoring* (<http://service.sap.com/monitoring>). Wählen Sie *Technical Information*, und wählen in der Liste das Dokument *Herstellung eines Data Supplier für den Alert-Monitor*.
- [MonSeg]** Das Monitorsegment der CCMS-Monitoring-Architektur
- Starten sie den *SAP Service Marketplace*, und springen Sie in den Bereich *System Management and Alert Monitoring* (<http://service.sap.com/monitoring>). Wählen Sie *Technical Information*, und wählen in der Liste das Dokument *Das Monitorsegment der CCMS-Monitoring-Architektur*.
- [RFC]** Online-Dokumentation zu RFC
- Rufen Sie hierzu die SAP-Bibliothek auf, und wählen Sie *mySAP-Technology-Komponenten* → *SAP Web Application Server* → *ABAP-Programmierung und Laufzeitumgebung (BC-ABA)* → *Externe Programmierschnittstellen* → *RFC-Programmierung in ABAP*.